Evaluating the Quality of Opponent Models in Automated Bilateral Negotiations

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

M.J.C. Hendrikx
born in Bergen op Zoom, the Netherlands

Interactive Intelligence Group
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
http://ii.tudelft.nl/

Evaluating the Quality of Opponent Models in Automated Bilateral Negotiations

Author: M.J.C. Hendrikx
Student id: 1358294
Email: m.j.c.hendrikx@student.tudelft.nl

Abstract

Automated negotiation agents are agents that interact in an environment for the settlement of a mutual concern. An important factor influencing the performance of a negotiation agent is how it takes the opponent into account. The main challenge in this aspect, is that opponents typically hide private information to avoid exploitation. In such a setting, an opponent model can help by estimating the opponent's strategy or preference profile. This work contains the first recent survey of opponent models in automated negotiation. One of the main conclusions of this survey, is that currently there is no fair method to evaluate and compare the quality of a set of opponent models. Insight in the quality of an opponent model could lead to the development of a better model. In this work we focus on a specific type of opponent models which model the opponent's preferences. Based on a detailed analysis of the factors influencing the quality of this type of opponent model, we introduce and apply two fair measurement methods to quantify the performance gain relative to not using an opponent model and the accuracy of the model. Our contribution to the field of automated negotiation is threefold; first, we provide a comprehensive survey of opponent models; second, we introduce a method to isolate the components of a negotiation strategy; finally, we construct and apply two fair evaluation methods to quantify the quality of a set of opponent models which model the opponent's preferences. Taken together, this work structures the field of opponent models and provides insight in how to improve existing models.

Thesis Committee:

Chair: Prof.dr. C.M. Jonker
Supervisor: Dr. K.V. Hindriks
External supervisor: Dr. M.M. de Weerdt
Daily supervisor: Drs. T. Baarslag

Preface

As part of a course on artificial intelligence we developed an automated negotiation agent which participated in the International Automated Negotiating Agent Competition 2011 (ANAC 2011), a yearly international competition in which negotiation agents compete on a set of beforehand unknown domains. To our surprise, we were one of the eight teams out of the initial eighteen to enter the finals. As part of our placement, we were invited to present our work on the AAMAS 2011 conference in Taiwan, Taipei.

This competition sparked my interest in automated negotiation; therefore, I decided to write my thesis on the topic of opponent modeling. During my thesis I wanted to publish some papers about my work. Therefore, in consultation with my supervisors, my thesis is a combination of paper with an overarching storyline.

I would like to thank Tim Baarslag, Koen Hindriks, and Catholijn Jonker both for their feedback and support during my thesis as well as their effort as co-authors. I especially thank Tim Baarslag for the large amount of time he has invested in answering my many "small" questions. I thank Alex Dirkzwager for his efforts as co-author and joint developer of the BOA framework and our ANAC 2011 and ANAC 2012 agents. In addition, I thank the Universiteitsfonds Delft and the Interactive Intelligence Group of the Delft University of Technology for sponsoring my trip to the AAMAS 2011 and AAMAS 2012. Furthermore, I thank Bart Vastenhouw and Ruud de Jong for their enthusiasm in arranging the large number of computers required to run all the experiments.

Finally, I thank my parents and my brother for their support during my thesis.

M.J.C. Hendrikx
Delft, the Netherlands
August 16, 2012

Contents

Chapter 1

Introduction

Negotiation is a process in which agents interact in an environment to arrange for the settlement of a mutual concern. Various disciplines studied the topic of negotiation, including economics [23, 26], electronic commerce [9, 21], artificial intelligence [8, 14, 16, 17, 18, 29], game theory [4, 8, 14, 18, 23, 25, 28], and social psychology [27].

Traditionally, negotiation is a necessary, but time-consuming and expensive activity. This has lead to an interest to automate negotiation [3, 9, 14, 17], for example in the setting of e-commerce [5, 15, 21]. This interest is fueled by the promise of computer agents being able to negotiate on behalf of human negotiators, or even outperforming them [5, 12, 21, 24].

In an automated negotiation it is common that opponents are unwilling to share their preferences and especially their negotiation strategy to avoid exploitation [7, 26, 30]. In practice, restricted information about the opponent is one of the key limiting factors to successful automated negotiations [22]. However, while automated agents often do not explicitly share private information, this information is implicitly embedded in their actions. Learning techniques can exploit this information to create a model of the opponent, which can be used to enhance the effectiveness and efficiency of the negotiation process [11, 30].

A large number of opponent models exist in literature, and more are introduced each year. In the survey of opponent modeling discussed in Chapter 2 we identify six categories of opponent models. An important direction for future work for all six types of models is a fair method to evaluate their quality. Insight in the quality of a model leads to an understanding of its strengths and weaknesses, which could lead to the development of a better model.

While various authors evaluated opponent models using different methods [6, 7, 13, 19, 30], to our knowledge there exists no work comparing the quality of a large set of opponent models which estimate the opponent's preferences. Therefore, the main aim of this work is to introduce such a method.

A first step towards a fair measurement for quality, is to introduce a method to isolate and switch the opponent model of an existing strategy for another model. In this case, the performance of the same strategy with different opponent models can be compared. Towards this end, in Chapter 3 we introduce a method to decompose negotiation strategies in three components: the acceptance strategy, bidding strategy, and opponent model. Using this framework, the components derived from multiple agents can be combined to create a new negotiation strategy.

As a first application of the framework, we created a state of the art negotiation strategy which participated in the International Automated Negotiating Agent Competition 2012 (ANAC 2012). The negotiation agent discussed in Chapter 4 finished third and additionally had the highest performance on one of the two settings in which the agents were tested.

Finally, also based on the framework introduced in Chapter 3, in Chapter 5 we introduce two measurement methods to fairly evaluate the performance and accuracy of an opponent model. Note that designing such methods is not trivial, as the quality of an opponent model depends on the negotiation setting, and each model is influenced differently. To be specific, the setting should be taken into account to avoid a biased measurement. The application of the methods results in an understanding of the quality of state of the art opponent models and insight in how these models can be improved.

To summarize, in my thesis the following main research questions are addressed:

1. Which types of opponent models exist in literature; and for each type: what are the main directions for future work?

2. Can existing agents be decomposed in a small set of components? Furthermore, is it possible to design a framework in which components of different agents can be combined to create a new negotiation strategy?

3. How to apply the framework discussed above to create a negotiation strategy improving the state of the art?

4. How to fairly evaluate the quality of an opponent model of the opponent's preferences acknowledging the influence of the negotiation setting on the quality of a model? In particular, how do existing models compare in quality?

The remainder of this paper is organized as follows. Chapter 2 surveys existing literature on opponent models. In Chapter 3 a framework is introduced which can be used to combine the components of different negotiation strategies to create a new strategy. Chapter 4 discusses how this framework is used to develop a state of the art negotiation strategy which competed in the ANAC 2012. In Chapter 5 we introduce two measurement methods to evaluate the quality of an opponent model. In Chapter 6 we discuss lessons learned and directions for future work. Chapter A discusses the components implemented to run all experiments. Finally, Chapter B discusses the contribution of each author.

Chapter 2

A Survey of Opponent Models in Automated Bilateral Negotiation

While a large number of opponent models have been introduced in literature, there is no recent survey on opponent modeling in automated negotiation. To be specific, the latest survey most similar to our work was by Beam and Segev in 1997 [3]. In this chapter we introduce a survey of the state of the art of opponent modeling. After finishing my thesis, we plan to submit the survey to a journal.

One of the main conclusions of this survey, is that there are currently no benchmarks to fairly evaluate the quality of an opponent model. Besides the advantage that using a benchmark the best opponent model can be found, it also provides insight in how to create better opponent models. Therefore, the remainder of this thesis discusses how to design a fair measurement method for opponent models which estimate the opponent's preferences.

Noname manuscript No.
(will be inserted by the editor)

A Comprehensive Survey of Opponent Models in Automated Bilateral Negotiation

Tim Baarslag · Mark J.C. Hendrikx · Koen V. Hindriks · Catholijn M. Jonker

Received: date / Accepted: date

Abstract Automated negotiators are agents that arrange for the settlement of a mutual concern by interacting in a negotiation. For a negotiation to be efficient and effective, it is important to take the opponent into account when deciding on a proposal. This is a challenging task, as negotiation is typically an incomplete information game, where the agents initially do not know their opponent's preferences nor strategy. In such a setting, constructing an opponent model can help predict the opponent's preferences and/or strategy in order to reach a better agreement. There has been a large body of research that focuses on modeling the opponent in bilateral negotiations. Despite the maturity of the field, there is no recent survey of currently existing opponent models. This work aims to bridge the gap by providing a comprehensive survey of opponent models used in the setting of bilateral negotiation. Our survey contributes to the state of the art in six ways: we identify common learning techniques used in opponent modeling; we introduce a taxonomy of opponent models; we discuss how agents can benefit from using opponent models; we survey current applications of opponent models; we provide an overview of measures to evaluate the quality of a model; and finally, we provide directions for future research.

Keywords negotiation · software agents · opponent model · opponent modeling · machine learning · survey

Tim Baarslag
Delft University of Technology, The Netherlands
E-mail: T.Baarslag@tudelft.nl

Mark J.C. Hendrikx
Delft University of Technology, The Netherlands
E-mail: M.J.C.Hendrikx@ii.tudelft.nl

Koen V. Hindriks
Delft University of Technology, The Netherlands
E-mail: K.V.Hindriks@tudelft.nl

Catholijn M. Jonker
Delft University of Technology, The Netherlands
E-mail: C.M.Jonker@tudelft.nl

As we plan to publish this paper in a journal this year, the version of the paper in this thesis is limited to an abstract. The committee members graded the full paper. A full version of the paper can be requested by contacting the authors.

Chapter 3

Decoupling the Components of a Negotiation Strategy

A negotiation strategy generally consists of multiple components including the opponent model. To fairly compare the performance of two opponent models, it should be possible to switch the model of a negotiation agent without changing any of the other components.

Towards this end, we constructed the BOA framework: a framework which allows to combine the components of multiple strategies to create a new negotiation strategy. A complete negotiation strategy in the BOA framework consists of a bidding strategy, an opponent model, and an acceptance strategy. The paper was accepted by the ACAN 2012 workshop on the AAMAS.

Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies

Tim Baarslag
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
T.Baarslag@tudelft.nl

Koen Hindriks
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
K.V.Hindriks@tudelft.nl

Mark Hendrikx
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
M.J.C.Hendrikx@tudelft.nl

Alex Dirkzwager
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
A.Dirkzwager@ii.tudelft.nl

Catholijn M. Jonker
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
C.M.Jonker@tudelft.nl

ABSTRACT

Every year, automated negotiation agents are improving on various domains. However, given a set of automated negotiation agents, current methods allow to determine which strategy is best in terms of utility, but not so much the reason of success. In order to study the performance of the individual components of a negotiation strategy, we introduce an architecture that distinguishes three components which together constitute a negotiation strategy: the bidding strategy, the opponent model, and the acceptance strategy.

Our contribution to the field of bilateral negotiation is twofold: first, we show that existing state-of-the-art agents are compatible with this architecture by re-implementing them in the new framework; secondly, as an application of our architecture, we systematically explore the space of possible strategies by recombining different strategy components, resulting in negotiation strategies that improve upon the current state-of-the-art in automated negotiation.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*intelligent agents, multi-agent systems*

General Terms

Algorithms, Bargaining, Experimentation, Negotiation

Keywords

Automated bilateral negotiation, BOA agent framework, decoupled, component-based, bidding strategy, opponent model, acceptance conditions, acceptance criteria

1. INTRODUCTION

Recently, many new automated negotiation agents have been developed. There is now a large body of negotiation strategies available, and with the emergence of the International Automated Negotiating Agents Competition (ANAC) [2, 4], new strategies are generated on a yearly basis.

While methods exist to determine the best negotiation agent given a set of agents [2, 4], we still do not know which type of agent is most effective in general, and especially why. It is impossible to exhaustively search the large (in fact, infinite) space of negotiation strategies; therefore, there is a need for a systematic way of searching this space for effective candidates.

Many of the sophisticated agent strategies that currently exist are comprised of a fixed set of modules. Generally, a distinction is made between three different modules: one module that decides whether the opponent's bid is acceptable; one that decides which set of bids could be proposed next; and finally, one that tries to guess the opponent's preferences and takes this into account when selecting an offer to send out. The negotiation strategy is a result of the complex interaction between these components, of which the individual performance may vary significantly. For instance, an agent may contain a module that predicts the opponent's preferences very well, but the agent may still perform badly utility-wise because it concedes far too quickly.

This means that overall performance measures, such as average utility obtained in a tournament, make it hard to pinpoint which components of an agent work well. To date no efficient method exists to identify to which of the components the success of a negotiating agent can be attributed. Finding such a method would allow to develop better negotiation strategies, resulting in better agreements; the idea being that well-performing components together will constitute a well-performing agent.

To tackle this problem, we propose to analyze three components of the agent design separately. We show that most of the currently existing negotiating agents can be fitted into the so-called *BOA framework* by putting together three main components in a particular way; namely: the *Bidding strategy*, an *Opponent model*, and an *Acceptance strategy*. We support this claim by re-implementing, among others, the ANAC agents in our framework. Furthermore, we show that the BOA agents are equivalent in behavior and performance to their original counterparts.

The advantages of fitting agents into the BOA framework are threefold: first, it allows to study the behavior and performance of individual components; second, it allows to systematically explore the space of possible negotiation strategies; third, the identification of unique interacting components simplifies the creation of new negotiation strategies.

Finally, we demonstrate the value of our framework by assembling, using already existing components, new negotiating agents that perform better than the current state-of-the-art. This shows that the BOA framework can yield better performing agents by combining better performing components.

The remainder of this paper is organized as follows. Section 2 discusses related work. In Section 3 the BOA agent framework is introduced, and we outline a research agenda on how to employ it. Section 4 provides evidence that many of the currently existing agents fit into the BOA framework, and discusses challenges in decoupling existing negotiation strategies. In Section 4.2 we illustrate how to test for equivalence of the original agent and its decoupled version. Section 5 shows how the BOA framework allows us to combine best practices in current agent design, leading to new, more effective strategies. Finally, in Section 6 we discuss lessons learned and provide directions for future work.

2. RELATED WORK

Since this paper introduces a framework based on a theory of components, we have surveyed literature that investigates and evaluates such components. There are three categories of related work: literature detailing the architecture of the negotiation strategy of an agent; work that discusses and compares the performance of a component of a negotiation strategy; and finally, literature that explores and combines a set of negotiation strategies to find an optimal strategy.

2.1 Achitecture of Negotiation Strategies

To our knowledge, there is little work in literature describing, at a similar level of detail as our work, the generic components of a negotiation strategy architecture. For example, Bartolini et al. [5] and Dumas et al. [8] treat the negotiation strategy as a singular component. There are however some notable exceptions.

Jonker et al. [16] present an agent architecture for multi attribute negotiation, where each component represents a specific process within the behavior of the agent, e.g.: attribute evaluation, bid utility determination, utility planning, and attribute planning. In contrast to our work, Jonker et al. focus on tactics for finding a counter offer and do not discuss acceptance conditions. However, there are some similarities between the two architectures. For example, the utility planning and attribute planning component correspond to the bidding strategy component in our architecture.

Ashri et al. [1] introduce a general architecture for negotiation agents, discussing components that resemble our architecture; however, the negotiation strategy is described from a BDI-agent perspective (in terms of motivation and mental attitudes). Components such as a proposal evaluator and response generator resemble an acceptance condition and bidding strategy respectively.

Hindriks et al. [13] introduce a generic architecture for negotiation agents in combination with a negotiation system architecture. Parts of the agent architecture correspond to the architecture presented in this paper; however, their focus is primarily on how the agent framework can be integrated into a larger system architecture. In addition, Hindriks et al. treat the acceptance condition and bidding strategy as a singular component.

2.2 Components of Negotiation Strategy

Evaluation of the performance of components is important to gain understanding of the performance of a negotiation strategy.

Regarding acceptance conditions, Baarslag et al. [3] analyze the performance of a set of acceptance conditions. These acceptance conditions depend on parameters such as time, utility of previous or next bid, and utility thresholds.

The notion of opponent model as a component of a negotiation strategy has been discussed by various authors, however to our knowledge there is no work comparing the performance of various state-of-the-art opponent models. Recently, Hindriks et al. [15] introduced different quality measures for learning, based on the estimated preference profile and the actual preference profile, but this has not been put to practice yet. Different types of models exist in literature, including opponent models that estimate the reservation value [24], the (partial) preference profile [14], the opponent's acceptance of offers [20], and that predict the opponent's next move [7].

Our work focuses on opponent models which estimate the (partial) preference profile, because most existing implementations fit in this category; however, in principle, our framework can accommodate for modeling the opponent's strategy as well. Our framework also allows to determine and to compare the performance of different opponent models by separating the implementation of the opponent model from the rest of the negotiation agent.

Although we are not the first to identify the BOA components in a negotiation strategy, our approach seems to be unique in the sense that we vary these components of the strategies, thereby creating new negotiation strategies, and improving the state-of-the-art in doing so.

2.3 Negotiation Strategy Space Exploration

There are at least four main types of baseline bidding strategies to compare the performance of a bidding strategy against: time dependent [10, 11], resource dependent [10], behavior dependent [10], and zero intelligence strategies [12].

Faratin et al. [10] start by analyzing the performance of pure negotiation tactics on single issue domains in a bilateral negotiation setting. The decision function of the pure tactic is then treated as a component around which the full strategy is built. While they discuss how tactics can be linearly combined, the performance of linearly combined tactics are not analyzed (in contrast to Matos et al. [19]), as they note that the set of possible strategies is too large to explore.

Matos et al. [19] use a set of baseline negotiation strategies with varying parameters. The negotiation strategies are combined linearly and encoded as chromosomes after which they are utilized by a genetic algorithm to analyze the effectiveness of the strategies. The fitness of an agent is its score in a negotiation competition. This approach is limited to acceptance criteria that specify a utility interval of acceptable values, and hence does not take time into account; furthermore, the agents do not employ explicit opponent modeling.

Eymann [9] also uses genetic algorithms with more complex negotiating strategies, evolving six parameters that influence the bidding strategy. The genetic algorithm uses the current negotiation strategy of the agent and the opponent strategy with highest average income to create a new strategy, similar to other genetic algorithm approaches (see Beam and Segev [6] for a discussion of the application of genetic algorithms in automated negotiation). The genetic algorithm approach mainly treats the negotiation strategy optimization as a search problem in which the parameters of a small set of strategies is varied using genetic algorithms. In our approach, we analyze a more complex space of newly developed negotiation strategies, as our pool of surveyed negotiation strategies consists of strategies introduced in the ANAC competition [2, 4], as well as the strategies discussed by Faratin et al. [10]. Furthermore, each strategy consists of components that can have parameters themselves. Our contribution is to define and implement a framework that allows to easily vary all main components of a negotiating agent.

3. THE BOA AGENT FRAMEWORK

In the last decade, many different negotiation strategies have been introduced in the search for an effective, generic automated negotiator (see related work Section 2). Current work often focuses on optimizing the negotiation strategy as a whole. We propose to direct our attention to a component-based approach, especially now that we have access to a large repository of mutually comparable negotiation strategies due to ANAC. This approach has several advantages:

1. Given measures for the effectiveness of the individual components of a negotiation strategy, we are able to pinpoint the most promising components, which gives insight into the reasons for success of the strategy;

2. Focusing on the most effective components helps to systematically search the space of possible negotiation strategies by recombining them into new strategies.

We make a distinction between two types of components in the sections below: elements that are part of the agent's environment, and components that are part of the agent itself.

3.1 Negotiation Environment

We employ the same *negotiation environment* as in [2, 4, 18]; that is, we consider *bilateral*, *real time* automated negotiations, where the interaction between the two negotiating parties is regulated by the alternating-offers protocol [21]. The agents negotiate over a set of issues, as defined by the negotiation *domain*, which holds the information of possible bids, constraints, and the discount factor. The negotiation happens in real time, and the agents are required to reach an agreement (i.e., one of them has to accept) before the deadline is reached. The timing of acceptance is particularly important because the utility may be discounted, that is: the value of an agreement may decrease over time.

In addition to the domain, both parties also have privately-known preferences described by their *preference profiles*. While the domain is common knowledge, the preference profile of each player is private information. This means that each player only has access to its own utility function, and does not know the preferences of its opponent. The player can attempt to learn this during the negotiation encounter by analyzing the *bidding history*, using an opponent modeling technique.

3.2 The BOA Agent

Based on a survey of literature and the implementations of currently existing negotiation agents, we identified three main components of a general negotiation strategy: a *bidding strategy*, possibly an *opponent model*, and an *acceptance strategy* (BOA). The elements of a BOA agent are visualized in Figure 1. In order to fit an agent into the BOA framework, it should be possible to discern these components in the agent design, with no dependencies between them. An exposition of the agents we considered is given in the next section, which will further motivate the choices made below.

1. **Bidding strategy**. A bidding strategy is a mapping which maps a negotiation trace to a bid. The bidding strategy can interact with the opponent model by consulting with it, passing one or multiple bids and see how they compare within the estimated opponent's utility space.
   Input: *opponent utility of bids, negotiation trace*.
   Output: *provisional upcoming bid*.

2. **Opponent model**. An opponent model is a learning technique that constructs a model of the opponent's negotiation profile. In our approach, the opponent model should be able to estimate the opponent's utility of a given bid.
   Input: *set of possible bids, negotiation trace*.
   Output: *estimated opponent utility of a set of bids*.

3. **Acceptance strategy**. The acceptance strategy determines whether the bid that the opponent presents is acceptable.
   Input: *provisional upcoming bid, negotiation trace*.
   Output: *send accept, or send out the upcoming bid*.

The components interact in the following way (the full process is visualized in Figure 1). When receiving an opponent bid, the BOA agent first updates the *bidding history* and *opponent model* to make sure the most up-to-date data is used, maximizing the information known about the environment and opponent.

Given the opponent bid, the *bidding strategy* determines the counter offer by first generating a set of bids with a similar preference for the agent. The *bidding strategy* uses the *opponent model* (if present) to select a bid from this set by taking the opponent's utility into account.

Finally, the *acceptance strategy* decides whether the opponent's action should be accepted. At first glance, it may seem counter-intuitive to make this decision *at the end* of the agent's deliberation cycle. Clearly, deciding upon acceptance *at the beginning* would have the advantage of not wasting resources on generating an offer that might never be sent out.

However, generating an offer first allows us to employ acceptance conditions that depend on the utility of the counter bid that is ready to be sent out. This method is widely used in existing agents [3]. Such acceptance mechanisms can make a more informed decision by postponing their decision on acceptance until the last step; therefore, and given our aim to incorporate as many agent designs as possible, we adopt this approach in our framework.

If the opponent's bid is not accepted by the acceptance strategy, then the bid generated by the bidding strategy is offered instead.

3.3 Employing the BOA framework

The component-based approach as outlined above enables us to follow at least two approaches: first of all, it allows us to independently analyze the components of every negotiation strategy that fits in to our framework. For example, by re-implementing the ANAC agents in the BOA framework, it becomes possible to compare the accuracy of all ANAC opponent models, and to pinpoint the best opponent model among them. Naturally, this helps to build better agents in the future.

Secondly, we can proceed to *mix* different BOA components, e.g.: replace the opponent model of the runner-up of ANAC by a different opponent model and test whether this makes a difference in placement. Such a procedure enables us to assess the reasons for success of an agent, and makes it possible to systematically search for an effective automated negotiator.

The first part of the approach gives insight in what components are best in isolation; the second part gives us understanding of their influence on the agent as a whole. At the same time, both approaches raise some key theoretical questions, such as:

1. Can the BOA components be identified in all, or at least most, current negotiating agents?

2. How do we measure the performance of the single components? Can a single best component be identified, or does this strongly depend on the other components?

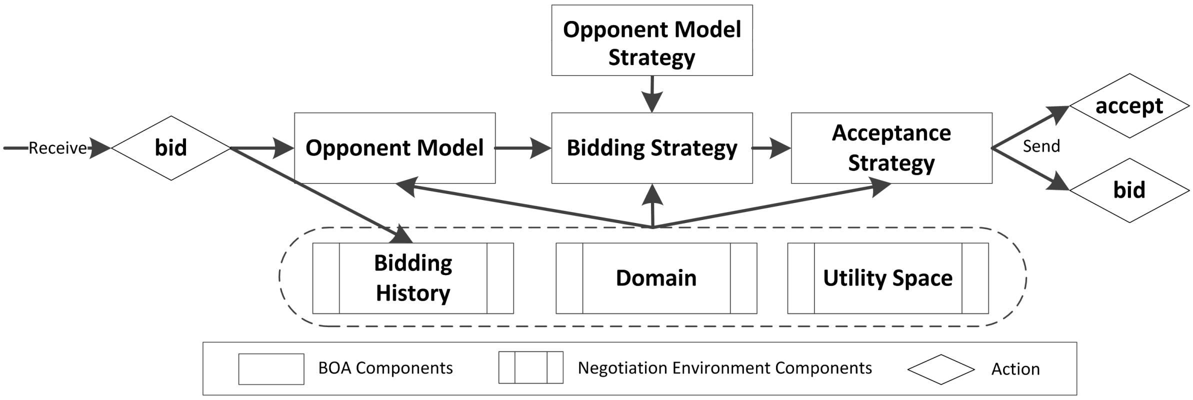

Figure 1: The BOA framework negotiation flow.

3. If the individual components perform better than others (with respect to some performance measure), does combining them in an agent also improve the agent's performance?

In this work we do not aim to fully answer all of the above questions; instead, we outline a research agenda, and introduce the BOA framework as a tool that can be used towards answering these questions.

Nonetheless, in the next section, we will provide empirical support for an affirmative answer to the first theoretical question: indeed, in many cases the components of the BOA framework can be identified in current agents, and we will also provide reasons for when this is not the case.

The answer to the second question depends on the component under consideration: for an opponent model, it is straightforward to measure its effectiveness: the closer the opponent model is to the actual profile of the opponent, the better it is. The quality of approximation can be measured in multiple ways [15], and should be balanced against other measures that also influence its performance. For instance, in a real time negotiation there is a trade-off between the required computational resources and expected quality of the opponent model.

The performance of the other two components of the BOA framework is better measured in terms of utility obtained in negotiation (as has been done for acceptance strategies in [3]), as there seems no clear alternative method to define the effectiveness of the acceptance strategy or bidding strategy in isolation. In any case, the BOA framework can be used as a research tool to help answer such theoretical questions.

Regarding the third question: suppose we take the best performing bidding strategy, equip it with the most faithful opponent model, and combine this with the most effective acceptance strategy; it would seem reasonable to assume this combination results in an effective negotiator. We plan to elaborate on this conjecture in future work (see also Section 6); however, Section 5 will already provide a first step towards this goal by recombining components of the ANAC 2011 agents to create more effective agents than the original versions.

4. DECOUPLING EXISTING AGENTS

In this section we provide empirical evidence that many of the currently existing agents can be decoupled by separating the components of a set of state-of-the-art agents. This section serves three goals: first, we discuss how existing agents can be decoupled in a bidding strategy, acceptance strategy, and possibly an opponent model; second, we argue that the BOA framework design is appropriate, as most agents will turn out to fit in our framework; third, we discuss a method to determine if the sum of the components – the BOA agent – is equal in behavior to the original agent.

4.1 Identifying the Components

In this section we identify the components of seventeen negotiating agents, taken from the ANAC competitions of 2010 [4], and 2011 [2], and of baseline strategies such as the time dependent agents [10, 11], and zero intelligence strategies [12]. We have selected these strategies as they are well-known and/or represent the current state-of-the-art in automated negotiation, having been implemented by various negotiation experts.

Since the agents were not designed with decoupling in mind, all agents had to be re-implemented to be supported by the BOA framework. Our decoupling methodology was to adapt an agent's algorithm to enable it to switch its components, without changing the agent's functionality. A method call to specific functionality, such as code specifying when to accept, was replaced by a more generic call to the acceptance mechanism, which can then be swapped at will. The contract of the generic calls are defined by the expected input and output of every component, as outlined in Section 3.2.

The first step in decoupling an agent is to determine which components can be identified. For example, in the ANAC 2010 agent *FSEGA* [22], an acceptance condition, a bidding strategy, and an opponent model can all be identified. The acceptance strategy combines simple, utility–based criteria (called $\mathbf{AC}_{\text{const}}$ and $\mathbf{AC}_{\text{prev}}$ in [3]) and can be easily decoupled in our framework. The opponent model is a variant of the *Bayesian opponent model* [14, 24], which is used to optimize the opponent utility of a bid. Since this usage is consistent with our framework (i.e., the opponent model provides opponent utility information), the model can be replaced by a generic opponent model interface. The final step is to change the bidding strategy to use the generic opponent model instead of specifically its own model. Other agents can be decoupled using a similar process.

Unfortunately, some agent implementations contained slight dependencies between different components. These dependencies needed to be resolved to separate the design into singular components. For example, both the acceptance strategy and bidding strategy of the ANAC 2011 agent *The Negotiator*[1] rely on a shared

[1]Descriptions of all ANAC 2011 agents can be found in [2].

target utility. In such cases, the agent can be decoupled by introducing Shared Agent State (SAS) classes. A SAS class avoids code duplication, and therefore performance loss, by containing the code that is shared between the components. One of the components uses the SAS to calculate the values of the required parameters and saves the results, while the other component simply asks for the saved results instead of duplicating the calculation.

Table 1 provides an overview of all agents we re-implemented in our framework, and more specifically, which components we were able to decouple. In fact, we were able to decouple all ANAC2011 and ANAC2011 agents except for *ValueModelAgent*. While *ValueModelAgent* can be theoretically decoupled, the strong coupling between its components results in too computationally heavy components when used separately.

As is evident from Table 1, the only possible obstacle in decoupling an agent is its usage of the opponent model. An agent's opponent model can be employed in multiple ways. Some agents, such as *Nice Tit for Tat*, attempt to estimate the Nash point on the Pareto frontier. Other common applications include: ranking a set of bids according to the opponent utility, reciprocating in opponent utility, and extrapolating opponent utility. The generic opponent model interface needs to sufficiently accommodate such requirements from the bidding strategy to make interchangeability possible. For this reason we require the opponent model interface to be able to produce the estimated opponent utility of an arbitrary negotiation outcome.

With regard to the opponent model, there are three groups of agents: first, there are agents such as *FSEGA* [22], which use an opponent model that can be freely interchanged; second, there are agents such as the ANAC 2010 winner *Agent K* [17], which do not have an opponent model themselves, but can be extended to use one. Such agents typically employ a bidding strategy that first decides upon a specific target utility range, and then picks a *random* bid within that range. These agents can easily be fitted with an opponent model instead, by passing the utility range through the opponent model before sending out the bid. Lastly, there are agents that cannot use an opponent model in any meaningful way, such as *Random Walker* [12], and there are agents such as *Gahboninho* and *BRAM Agent* that use a frequency-based opponent model which is not compatible with our framework, as their opponent models do not yield enough information to compute the opponent utility of bids. For this type of agent, we consider the opponent model part of the bidding strategy.

When decoupling the agents, it becomes apparent that the bidding strategy component varies greatly between different agents. In contrast, there are only two main types of opponent models being used: Bayesian models and frequency models. Bayesian models are an implementation of a (scalable) model of the opponent preferences that is updated using Bayesian learning [14, 24].

The main characteristic of frequency based models is that they track the frequency of occurrence of issues and values in the opponent's bids and use this information to estimate the opponent's preferences. In practice, Bayesian models are more computationally intensive, whereas frequency models are relatively light-weight.

After comparing the different implementation variants in all agents, we consider the Bayesian model of *IAMhaggler 2010* and the frequency model of *HardHeaded* representative of their type, as we believe that both are the most accurate and computationally efficient implementations.

Similar to the opponent models, most agents use variations and combinations of a small set of acceptance conditions. Specifically, many agents use simple thresholds for deciding when to accept (called $\mathbf{AC}_{\text{const}}$ in [3]) and linear functions that depend on the utility of the bid under consideration ($\mathbf{AC}_{\text{next}}(\alpha, \beta)$ [3]).

Agent	B	O	A
Agent K	✓	∅	✓
Agent K2	✓	∅	✓
Agent Smith	✓	✓	✓
BRAM Agent	✓	–	✓
FSEGA	✓	✓	✓
Gahboninho	✓	–	✓
HardHeaded	✓	✓	✓
IAMcrazyHaggler	✓	∅	✓
IAMhaggler2010	✓	✓	✓
IAMhaggler2011	✓	∅	✓
Nice Tit for Tat	✓	✓	✓
Nozomi	✓	∅	✓
Offer Decreasing	✓	–	✓
Random Walker	✓	–	✓
TheNegotiator	✓	∅	✓
Time dependent agent	✓	∅	✓
Yushu	✓	∅	✓

Table 1: Overview of components identified in every agent. ✓: original has component that can be decoupled. ∅: original has no component, but it can be added. – : no support for such a component.

4.2 Testing Equivalence of BOA Agents

A BOA agent should behave identically to the agent from which its components are derived. Equivalence can be verified in two ways; first, given the same negotiation environment and the same state, both agents should behave in exactly identical ways (Section 4.2.1); second, the performance in a real time negotiation of both agents should be similar (Section 4.2.2).

4.2.1 Identical Behavior Test

Two deterministic agents can be considered equivalent if they perform the same action given the same negotiation trace. There are two main problems in determining equivalence: first, most agents are nondeterministic, as they behave randomly in certain circumstances; for example, when picking from a set of bids of similar utility; second, the default protocol in GENIUS uses real time [18], which is highly influenced by cpu performance. This entails that in practice, two runs of the same negotiation are never exactly equivalent.

To be able to run an equivalence test despite of the agents choosing random actions, we fixed the seeds of the random functions of the agents. The challenge of working in real time was dealt with by changing the real time deadline to a maximum amount of rounds. Since time does not pass within a round, cpu performance does not play a role.

All agents were evaluated on the ANAC2011 domains (see [2] for a domain analysis). The ANAC2011 domains vary widely in characteristics: the number of issues ranges from 1 to 8, the size from 3 bids to 390.625 bids, and the discount from none (1.0) to strong (0.424). Some ANAC2010 agents, specifically *Agent Smith* and *Yushu*, were not designed for large domains and were therefore run on a subset of the domains.

The *opponent strategies* used in the identical behavior test should satisfy two properties: the opponent strategy should be deterministic, and secondly, the opponent strategy should not be the first to accept, to avoid masking errors in the agent's acceptance strategy.

Given these two criteria, we used the standard time-dependent tactics [10, 11] for the opponent bidding strategy. Specifically, we use *Hardliner* ($e = 0$), *Linear Conceder* ($e = 1$), and *Conceder* ($e = 2$). In addition, we use the *Offer Decreasing* agent which offers the set of all possible bids in decreasing order of utility.

All original and BOA agents were evaluated against all four opponents on eight domains, using both preference profiles defined on each domain. An agent running both strategies in parallel was used to check that both strategies were equivalent.

After the experiments were performed, the test results indicated that all BOA agents were exactly identical to their original counterparts.

4.2.2 Similar Performance Test

Two agents can perform the same action given the same input, but may still achieve different results because of differences in their real time performance. When decoupling agents, there is a trade-off between performance and interchangeability of components. For example, most agents record only a partial negotiation history, while some acceptance strategies require the full history of the agent and/or its opponent. In such cases, the agent can be constrained to be incompatible with these acceptance strategies, or generalized to work with the full set of available acceptance strategies. We typically elected the most universal approach, even when this negatively influenced performance. We will demonstrate that while there is some performance loss when decoupling existing agents, it does not significantly impact the negotiation outcome.

The performance of the BOA agents was tested by letting them participate in the ANAC 2011 tournament (using the same setup, cf. [2]). The decoupled ANAC 2011 agents replaced the original agents, resulting in an 8×8 tournament, while the ANAC 2010 agents were added to the tournament, resulting in 9×9 tournaments.

For our experimental setup we used computers that were slower compared to the IRIDIS high-performance computing cluster that was used to run ANAC 2011. As we were therefore unable to reproduce exactly the same data, we first recreated our own ANAC 2011 tournament data (Appendix B), which is used as our baseline to benchmark the decoupled agents. The difference in performance caused small changes compared to the official ANAC 2011 ranking, as *Agent K2* moved up from 5th to 3rd place.

Table 2 in Appendix A provides an overview of the results. We evaluated the performance in terms of time of agreement and average overall utility. From these results, we can conclude that the variation is minimal: the largest difference between the original and decoupled agents is 0.010 for the average time of agreement (due to *Agent Smith*) and 0.009 for the average utility (due to *HardHeaded*). Therefore the BOA agents and their original counterparts show comparable performance.

5. APPLICATIONS OF THE BOA FRAMEWORK

The BOA framework can be used to compare the performance of components and, using this knowledge, we can search for negotiation strategies that improve the current state-of-the-art. In this section we discuss a first exploratory test setup in which we change the acceptance condition and opponent model of existing ANAC 2011 agents to improve their performance.

Despite the reduced negotiation space that is searched, the space still needs to be scaled down. Decoupling n agents can in theory give rise to n^3 new agents if each agent implements all BOA components (see Figure 1), and even larger if we allow different parameters for each component. In practice, it quickly becomes unfeasible to search the full Cartesian product of components. To reduce the space, we have devised a method to test multiple acceptance criteria at the same time, as is explained below.

5.1 Scaling the Negotiation Space

Suppose that two negotiating BOA agents A and B have identical bidding mechanisms and the same opponent modeling technique, so that only their acceptance criteria differs. Furthermore, suppose agent A accepts in the middle of the negotiation, and agent B at the end. The agents have accepted at a different time during the negotiation, but the bidding behavior will be identical until the acceptance occurred. The only difference between the complete traces is that the trace of agent A is cut-off in the middle of the negotiation.

In the BOA framework we exploit this property by running all acceptance conditions in parallel, and recording when each acceptance condition accepts. This reduces the amount of component combinations from n^3 to n^2 as the n acceptance conditions are reduced to 1. This approach is from now on referred to as multi-acceptance criteria (MAC).

In addition, since we support parameters for the acceptance conditions, a large number of acceptance conditions varying only in the value of their parameters can be tested during the same negotiation thread. Note that this approach assumes that checking additional acceptance conditions does not introduce a large computational overhead. In practice we found that the computational overhead was less than 5%, even when more than 50 variants of acceptance conditions were used at the same time.

Note that a similar technique cannot be applied for the bidding strategy and the opponent model, as both components directly influence the negotiation trace.

Even if MAC is applied, there are still n^2 possible combinations to explore. This is already problematic for a limited amount of domains and agents. To illustrate, ANAC 2011 consists of 448 negotiation sessions [2] which may all last 3 minutes. In worst case, it requires 22 hours to run a single tournament, and almost four weeks for running it 28 times, as we did for the similarity test discussed in Section 4.2.2. Towards improving scalability, we extended GENIUS so that a negotiation tournament may be distributed among multiple computers.

5.2 Improving the State-of-the-Art

Using the scaling methods discussed in the previous section, we were able to explore a reduced space of negotiation strategies. We opted to limit our attention to the ANAC 2011 agents for two reasons: first, because it is a competition that already has verified data which can be re-used; second, the ANAC 2011 tournament is the most recent incarnation of ANAC at the time of writing, and can therefore be assumed to contain state-of-the-art negotiation agents.

5.2.1 Searching the Negotiation Space

For each agent of our test setup, the original bidding strategy was fitted with alternative acceptance conditions and opponent models. We used the following sets of BOA components:

- **(B)** For the bidding strategies, the seven decoupled agents from ANAC 2011 were used (see Table 1).
- **(O)** As our opponent model set we elected the two representative opponent models we identified in Section 4.1 (i.e., a Bayesian model and a frequency model). In addition, we allowed the strategies to use *no* opponent model, as the computational overhead of an opponent model could lead to worse performance.

(A) All acceptance conditions of the seven decoupled agents of the ANAC 2011 were used, except for the acceptance condition of *Gahboninho*, as it is relatively heavy-weight, resulting in poor cpu performance.

In addition to the existing acceptance mechanism components, we used acceptance mechanisms that combined certain acceptance criteria, such as $\mathbf{AC}_{\text{combi}}(T, \text{MAX}^W)$ [3], and the discounted version of $\mathbf{AC}_{\text{next}}$, called $\mathbf{AC}_{\text{next}}^{\text{disc}}(\alpha, \beta, \gamma, \delta)$. Similar to the acceptance condition of *IAMcrazyHaggler* [23], it differentiates between domains with and without discount factors; on undiscounted domains, it behaves identically to $\mathbf{AC}_{\text{next}}(\alpha, \beta)$ [3]; on the discounted domains it is equal to $\mathbf{AC}_{\text{next}}(\gamma, \delta)$. Table 4 in Appendix C provides an overview of all 83 tested acceptance conditions.

All possible combinations were run three times during an exploratory search to determine the best combination of components for each agent. Similar to the equivalence test, we replaced the original agent strategy by the new strategy and measured its performance in the ANAC 2011 tournament. The best agent strategies were run 10 times to determine whether the average utility was significantly improved.

5.2.2 Results

From the seven agents analyzed in the test set we were able to considerably improve four: *Gahboninho*, *Agent K2*, *Nice Tit for Tat*, and *BRAM Agent*. All four perform significantly better than their original (*two-tailed t-test*, $p < 0.01$).

For the other three agents, all the combinations of acceptance conditions and opponent models resulted in similar or worse performance. This indicates that the components of these strategies are well geared to one another. Note that this does not mean that the match is optimal, it only indicates that the components of the strategy are optimal within the tested set of components. Table 5 in Appendix D provides an overview of the performance of the best combination of components for each agent.

We note that using an improved opponent model does not ensure a better negotiation outcome, and in some cases can even result in worse performance due to the overhead caused by updating the model. An interesting direction for future work could be to quantify the contribution of opponent models to the performance of the ANAC agents.

All in all, the results demonstrate that the BOA framework not only assists in exploring the negotiation strategy space and improving existing agents, but it also helps to identify which components of the agent are decisive in its performance.

6. CONCLUSION AND FUTURE WORK

This paper introduces a framework that distinguishes the bidding strategy, the opponent model, and the acceptance strategy in automated negotiation strategies and recombines these components to systematically explore the space of automated negotiation strategies. The main idea behind the BOA framework is that we can identify several components in a negotiating agent, all of which can be optimized individually. Our motivation in the end is to create a proficient negotiating agent by combining the best components.

We have shown that many of the existing negotiation strategies can be re-fitted into our framework. We identified and classified the key components in them, and we have demonstrated that the original agents and their decoupled versions have identical behavior and similar performance. Finally, we have given an application of the BOA framework by recombining different components of the ANAC agents, and we have demonstrated this can significantly improve their performance.

One obvious direction of future research is to look at different BOA components in isolation; for example, to find the best opponent model that is currently available. After identifying the best performing components, we can turn our attention to answer whether combining effective components leads to better overall results, and whether an optimally performing agent can be created by taking the best of every component. Our framework allows us to make these questions precise and provides a tool for answering these questions.

Another possible improvement is extend the focus of current work on preference profile modeling techniques to a larger class of opponent modeling techniques, such as strategy prediction. Also, an agent is currently equipped with a single component during the entire negotiation session. It would be interesting to run multiple BOA components in parallel, and use recommendation systems to elect the best component at any given time.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075.

7. REFERENCES

[1] R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In *Proceedings of the 3rd Central and Eastern European conference on Multi-agent systems*, pages 136–146. Springer-Verlag, 2003.

[2] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence Journal*, Submitted.

[3] Tim Baarslag, Koen Hindriks, and Catholijn Jonker. Acceptance conditions in automated negotiation. In *Proceedings of The Forth International Workshop on Agent-based Complex Automated Negotiations (ACAN 2011)*, 2011.

[4] Tim Baarslag, Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The first automated negotiating agents competition (ANAC 2010). In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence*, pages 113–135, Berlin, Heidelberg, 2012. Springer-Verlag.

[5] C. Bartolini, C. Preist, and N.R. Jennings. A generic software framework for automated negotiation. In *First International Conference on Autonomous Agent and Multi-Agent Systems*. Citeseer, 2002.

[6] C. Beam and A. Segev. Automated negotiations: A survey of the state of the art. *Wirtschaftsinformatik*, 39(3):263–268, 1997.

[7] R. Carbonneau, G.E. Kersten, and R.klaue2001automated Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2):1266–1273, 2008.

[8] M. Dumas, G. Governatori, A.H.M. Ter Hofstede, and P. Oaks. A formal approach to negotiating agents

development. *Electronic Commerce Research and Applications*, 1(2):193–207, 2002.

[9] T. Eymann. Co-evolution of bargaining strategies in a decentralized multi-agent system. In *AAAI fall 2001 symposium on negotiation methods for autonomous cooperative systems*, pages 126–134, 2001.

[10] P. Faratin, C. Sierra, and N.R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3):159–182, 1998.

[11] S.S. Fatima, M. Wooldridge, and N.R. Jennings. Multi-issue negotiation under time constraints. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 143–150. ACM, 2002.

[12] D.K. Gode and S. Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, pages 119–137, 1993.

[13] K. Hindriks, C. Jonker, and D. Tykhonov. Towards an open negotiation architecture for heterogeneous agents. *Cooperative Information Agents XII*, pages 264–279, 2008.

[14] K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, volume 1, pages 331–338. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[15] K.V. Hindriks and D. Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis*, pages 46–59, 2010.

[16] C.M. Jonker, V. Robu, and J. Treur. An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.

[17] Shogo Kawaguchi, Katsuhide Fujita, and Takayuki Ito. Compromising strategy based on estimated maximum utility for automated negotiation agents competition (ANAC-10). In Kishan G. Mehrotra, Chilukuri K. Mohan, Jae C. Oh, Pramod K. Varshney, and Moonis Ali, editors, *IEA/AIE (2)*, volume 6704 of *Lecture Notes in Computer Science*, pages 501–510. Springer, 2011.

[18] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 2012.

[19] N. Matos, C. Sierra, and N.R. Jennings. Determining successful negotiation strategies: An evolutionary approach. In *Proceedings of the International Conference on Multi Agent Systems*, pages 182–189. IEEE, 1998.

[20] Y. Oshrat, R. Lin, and S. Kraus. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 377–384. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[21] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–109, 1982.

[22] Liviu Dan Serban, Gheorghe Cosmin Silaghi, and Cristian Marius Litan. Agent fsega - time constrained reasoning model for bilateral multi-issue negotiations. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence*, pages 159–165, Berlin, Heidelberg, 2012. Springer-Verlag.

[23] C.R. Williams, V. Robu, E.H. Gerding, and N.R. Jennings. Iamhaggler: A negotiation agent for complex environments. 2010.

[24] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human-Computers Studies*, 48(1):125–141, 1998.

APPENDIX

A. SIMILARITY TEST

	Avg. time of agr.	SD time of agr.	Avg. utility	SD utility
Agent K (Org.)	0.713	0.0057	0.666	0.0035
Agent K (Dec.)	0.714	0.0061	0.672	0.0045
Agent Smith (Org.)	0.469	0.0083	0.703	0.0041
Agent Smith (Dec.)	0.479	0.0053	0.707	0.0041
FSEGA (Org.)	0.425	0.0013	0.721	0.0009
FSEGA (Dec.)	0.426	0.0041	0.721	0.0024
IAMcrazyHaggler (Org.)	0.591	0.0103	0.699	0.0078
IAMcrazyHaggler (Dec.)	0.587	0.0069	0.702	0.0099
IAMhaggler2010 (Org.)	0.633	0.0110	0.682	0.0093
IAMhaggler2010 (Dec.)	0.636	0.0101	0.684	0.0066
Nozomi (Org.)	0.663	0.0071	0.704	0.0063
Nozomi (Dec.)	0.666	0.0062	0.708	0.0053
Yushu (Org.)	0.798	0.0030	0.715	0.0035
Yushu (Dec.)	0.800	0.0026	0.717	0.0037
Agent K2 (Org.)	0.619	0.0046	0.685	0.0040
Agent K2 (Dec.)	0.621	0.0050	0.686	0.0034
BRAM Agent (Org.)	0.683	0.0089	0.683	0.0054
BRAM Agent (Dec.)	0.687	0.0060	0.683	0.0033
Gahboninho (Org.)	0.667	0.0055	0.736	0.0044
Gahboninho (Dec.)	0.668	0.0053	0.742	0.0015
HardHeaded (Org.)	0.738	0.0009	0.758	0.0024
HardHeaded (Dec.)	0.735	0.0028	0.749	0.0034
IAMhaggler2011 (Org.)	0.494	0.0102	0.685	0.0023
IAMhaggler2011 (Dec.)	0.493	0.0078	0.683	0.0024
Nice Tit for Tat (Org.)	0.677	0.0078	0.676	0.0043
Nice Tit for Tat (Dec.)	0.683	0.0070	0.668	0.0025
The Negotiator (Org.)	0.716	0.0016	0.679	0.0027
The Negotiator (Dec.)	0.716	0.0014	0.679	0.0023

Table 2: The table shows performance (with standard deviation) of agents in an ANAC 2011 tournament before and after being decoupled.

B. RESULTS OF ANAC COMPETITION

Agent	Amsterdam Trip	Camera	Car	Energy	Grocery	Company Acquisition	Laptop	Nice or Die	Mean utility
HardHeaded	0.891	**0.818**	**0.961**	0.664	0.725	0.747	0.683	**0.571**	0.757
Gahboninho	**0.912**	0.659	0.928	**0.681**	0.667	0.744	0.726	**0.571**	0.736
Agent K2	0.759	0.719	0.922	0.467	0.705	0.777	0.703	0.429	0.685
IAMhaggler 2011	0.769	0.724	0.873	0.522	0.725	**0.814**	**0.749**	0.300	0.685
BRAM Agent	0.793	0.737	0.815	0.420	0.724	0.744	0.661	**0.571**	0.683
The Negotiator	0.792	0.744	0.913	0.524	0.716	0.748	0.674	0.320	0.679
Nice Tit for Tat	0.733	0.765	0.796	0.508	0.759	0.767	0.660	0.420	0.676
Value Model Agent	0.839	0.778	0.935	0.012	**0.767**	0.762	0.661	0.137	0.611

Table 3: ANAC 2011 results of our hardware (n = 10).

C. VARIABLES USED FOR ACCEPTANCE CONDITIONS

Acceptance Condition	Range	Increments
$\mathbf{AC}_{\text{maxinwindow}}(T)$	T$\in$ [0.95, 0.99]	0.01
$\mathbf{AC}^{\text{disc}}_{\text{next}}(\alpha,\beta,\gamma,\delta)$	$\alpha \in$ [1.0, 1.05]	0.05
	$\beta \in$ [0.0, 0.1]	0.05
	$\gamma \in$ [1.0, 1.1]	0.05
	$\delta \in$ [0.0, 0.15]	0.05
$\mathbf{AC}_{\text{HardHeaded}}$	–	–
$\mathbf{AC}_{\text{TheNegotiator}}$	–	–
$\mathbf{AC}_{\text{NiceTitForTat}}$	–	–
$\mathbf{AC}_{\text{BRAMAgent}}$	–	–
$\mathbf{AC}_{\text{AgentK2}}$	–	–
$\mathbf{AC}_{\text{IAMhaggler2011}}$	–	–

Table 4: Variables that were used for the acceptance conditions.

D. IMPROVED AGENTS STRATEGY RESULTS

Agent	Original Utility	Best Performing Opponent Model	Best Performing Acceptance Condition	Improved Utility
Gahboninho	0.736	Original Model (No Model)	$\mathbf{AC}^{\text{disc}}_{\text{next}}(\alpha,\beta,\gamma,\delta)$ α: 1.0; β: 0.0; γ: 1.1; δ: 0.1;	0.759
Agent K2	0.685	IAMhaggler Model	$\mathbf{AC}^{\text{disc}}_{\text{next}}(\alpha,\beta,\gamma,\delta)$ α: 1.0; β: 0.0; γ: 1.0; δ: 0.15;	0.724
BRAM Agent	0.683	Original Model (No Model)	$\mathbf{AC}^{\text{disc}}_{\text{next}}(\alpha,\beta,\gamma,\delta)$ α: 1.0; β: 0.05; γ: 1.1; δ: 0.1;	0.697
Nice Tit For Tat	0.676	Original Model (Bayesian Model)	$\mathbf{AC}_{\text{maxinwindow}}(t)$ t: 0.99	0.696
HardHeaded	0.757	Original Model (Frequency Model)	$\mathbf{AC}_{\text{HardHeaded}}$	–
TheNegotiator	0.679	Original Model (No Model)	$\mathbf{AC}_{\text{TheNegotiator}}$	–
IAMhaggler2011	0.685	Original Model (No Model)	$\mathbf{AC}_{\text{IAMhaggler2011}}$	–

Table 5: Results of the improved agent strategies in an ANAC 2011 tournament (for n = 10 runs). The first four agents were improved significantly. Ther other two agents did not improve significantly in our test setup.

Chapter 4

Applying the BOA Framework

In Chapter 3 the BOA framework was introduced which can be used to create a negotiation strategy by selecting a bidding strategy, opponent model, and acceptance strategy. This chapter discusses the design of a negotiation agent created using the BOA framework. The negotiation agent participated in the ANAC 2012 with sixteen other teams in which it won the award for the highest utility on undiscounted domains – one of the two parts of the competition – and overall the third place. Our team was the only team of master students to enter the finals.

This paper is part of the post-proceedings of the ACAN 2012 workshop which took place at the AAMAS 2012.

A Competitive Strategy for Real-time Bilateral Negotiations
Application of the BOA Framework

A.S.Y. Dirkzwager, M.J.C.Hendrikx

Abstract Each year the ANAC introduces an increasingly complex negotiation setting to stimulate the development of negotiation strategies. This year, the ANAC2012 competition introduced a real-time bilateral negotiation setting with reservation values and discounts. This work introduces the strategy of the third place finalist and agent with best performance in undiscounted domains: *TheNegotiator Reloaded* (TNR). TNR is the first ANAC agent created using the BOA framework, a framework allowing to separately develop and optimize different components of a negotiation strategy. The agent uses a time-based strategy of which the concession speed is determined based on an analysis of the opponent's behavior and discount factor. In addition, an opponent model is used to determine the maximum concession based on an estimation of the Kalai-Smorodinsky point and the reservation value of the domain. Our contribution to the field of bilateral negotiation is threefold: first, we present the implementation and optimization of a negotiation strategy for a complex negotiation setting; second, we implement and use a set of quality measures to analyze the agent's performance; finally, we discuss how the agent could be improved and extended.

1 Introduction

Last year, the ANAC2011 competition introduced a negotiation setting in which agents competed in a real-time bilateral negotiation on domains with time-based discounts. This year the setting was further extended to include reservation values. Interestingly, this setting is of sufficient complexity to capture some real-life negotiations, for example a negotiation about a used car. Introducing an advanced negotiation strategy for the ANAC2012 competition could help in improving the outcome and efficiency of current (human) negotiations.

This work introduces the strategy of the third place finalist and agent with the best performance on undiscounted domains in the ANAC2012 competition: *TheNegotiator Reloaded* (TNR). TNR is the first agent based on the BOA framework, a framework which allows to separately develop the bidding strategy, opponent model, and acceptance conditions and to easily swap these components [2]. The agent uses an analysis of the opponent's strategy and the discount of the domain to determine its concession speed. In addition, TNR estimates the Kalai-Smorodinsky point and uses the reservation value to determine its maximum concession.

A.S.Y. Dirkzwager · M. Hendrikx
Interactive Intelligence Group, Delft University of Technology, Mekelweg 4, Delft, The Netherlands,
e-mail: {A.S.Y.Dirkzwager,M.J.C.Hendrikx}@student.tudelft.nl

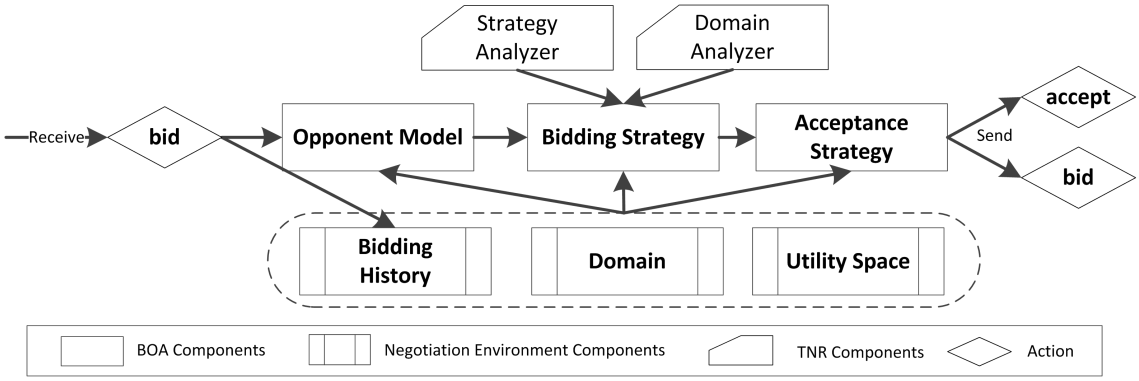

Fig. 1: Overview of the components of *TheNegotiator Reloaded.*

Towards stimulating the development of better performing negotiation strategies, this work discusses the implementation of TNR and analyzes its performance. Section 2 discusses the negotiation strategy and how it is implemented and optimized using the BOA framework. In Section 3 a toolkit of quality measures is used to quantify the performance of the negotiation strategy. Finally, Section 4 provides directions for future research.

2 Negotiation Strategy

This section discusses the strategy of *TheNegotiator Reloaded.* Section 2.1 briefly describes the BOA framework used to create TNR, for a more detailed description we refer to [2]. Following, Section 2.2 discusses how the BOA framework is used to implement TNR's main components.

2.1 Introduction to the BOA Framework

The BOA framework is a framework implemented for GENIUS which allows to separately develop the components of a negotiation strategy. The BOA framework makes a distinction between between three types of components: a **B**idding strategy which maps a negotiation trace to a bid; an **O**pponent model, which is a learning technique used to model the opponent's preference profile; and finally an **A**cceptance strategy which determines whether the opponent's offer is acceptable. A full negotiation strategy is created by selecting a component for each of the three types. In fact, the full Cartesian product of these components can be chosen resulting in a large space of possible negotiation strategies.

The advantages of implementing an agent as a set of BOA compatible components (a BOA agent) are threefold: first, it allows to study the performance of individual components; second, it allows to systematically explore the space of possible negotiation strategies to find an optimal strategy; third, the identification of unique interacting components simplifies the creation of new negotiation strategies.

Figure 1 provides an overview of how the components interact. When receiving an opponent bid, the BOA agent first updates the *bidding history* and *opponent model* to ensure that the most up-to-date data is used, maximizing the information known about the environment and opponent. Given the opponent bid, the *bidding strategy* determines the counter offer by first generating a set of bids with a similar utility for the agent. Following, the *bidding strategy* uses the *opponent model* to select a bid from this set by taking the opponent's utility into account. Finally, the *acceptance strategy* decides whether the opponent's action should be accepted. If the opponent's bid is not accepted by the *acceptance strategy*, then the bid generated by the *bidding strategy* is offered instead.

Each component of TNR was implemented separately using the BOA framework. The following section discusses the implementation and optimization of each component in detail.

2.2 Implementing the BOA Components

This section discusses the BOA components of TNR. Figure 1 provides an overview of the components of the agent's negotiation strategy. Section 2.2.1 discusses the bidding strategy. In Section 2.2.2 the opponent model is described. Finally, Section 2.2.3 discusses the acceptance strategy. Each section is subdivided into implementation and optimization.

2.2.1 Bidding Strategy

Implementing the Bidding Strategy

TNR is a BOA agent which uses a *domain analyzer* and *strategy analyzer* to adapt its decision function during the negotiation. TNR uses the standard time-dependent decision function discussed by Faratin et al. [4]. We specifically chose to adopt this decision function, as its parameters can easily be adapted during the negotiation. An overview of the bidding strategy is depicted in Figure 2.

The first step taken by TNR, is that it determines if the discount is low, medium, or high. Following, the time is divided in a set of windows. At the start of each window, the *domain analyzer* is used to estimate the Kalai-Smorodinsky point and the *strategy analyzer* is asked if the opponent is a conceder or hardliner. Note that preferably, these calculations should be done each round. Unfortunately, this is too computationally expensive and therefore we use a number of windows. The next step is to choose the parameters of the standard time-dependent decision function depicted in Equation 1.

$$P_{min} + (P_{max} - P_{min}) \cdot (1 - F(t)) \;\; where \;\; F(t) = k + (1-k) \cdot t^{1/e}. \tag{1}$$

The first step in selecting the parameters, is to select the concession speed e. The value for e is selected from a table which, given the discount type (low, medium, high) and the opponent's strategy type (conceder or hardliner), specifies a concession speed e. While the discount type does not change, the identification of the opponent's strategy can differ each window. The second and final step is to set the maximum concession P_{min} to the estimated Kalai-Smorodinsky point calculated by the *domain analyzer*.

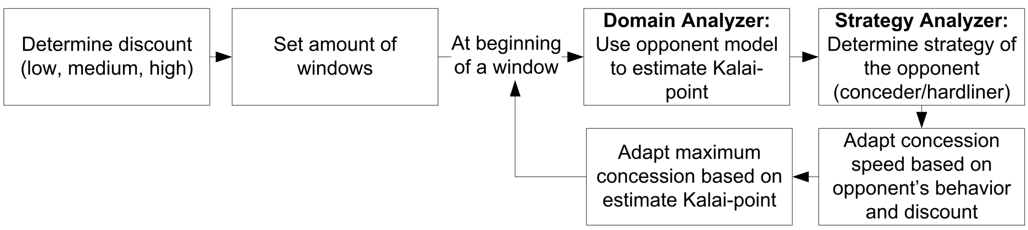

Fig. 2: Overview of bidding strategy of TNR.

For domains with a discount, P_{min} is multiplied by the discount to ensure that the agent concedes faster. Following, if the undiscounted reservation value is higher than P_{min}, then P_{min} is set to the reservation value. However, if the resulting P_{min} is smaller than 0.4, P_{min} is set to 0.4. The variable k is always zero, and P_{max} is always one.

The target utility is used by the bid selector to select a bid with a utility as close as possible to the target utility by using the binary search algorithm. Note that an alternative approach would be to select a set of bids and use an opponent model to select the best bid for the opponent. However, in our tests this approach did not result in a significant gain in performance.

Optimizing the Bidding Strategy

As discussed in the first part of this section, the concession speed e is chosen based on the opponent's strategy type (conceder or hardliner) and discount type (low, medium, high). Initially, we distinguished four different discount types: no [1.00], low [0.85, 1.00), medium (0.40 - 0.85) and high [0.0 - 0.40]. This distinction was made to allow our agent to be more flexible and perform better in a tournament where these different types of discounts can be present. To find the optimum values of these eight parameters, we created four variants of each ANAC2011 domain based on the discount types. The exact discount values were chosen randomly (within the discount type range) for each domain. Following a set of representative opponent's was defined. We chose to use the ANAC2011 agents except for *ValueModelAgent*. This agent was excluded both to decrease the test size, as well as an effort to decrease the standard deviation of the discounted utility. The *Energy* and *NiceOrDie* domains were also discarded for the same reasons.

For each domain type a competition was ran in which TNR competed against all ANAC2011 agents. Each domain type requires two parameters, corresponding to both possible strategy types of the opponent. Using trial-and-error the optimum value maximizing the discounted utility for each type of domain was found. The optimized values resulted in a first place for our agent in each of the four tournaments. Following, we further optimized the concession speeds to maximize the distance to the second place. After the optimization step, we noted that there was little difference between the domains with no and low discount. Therefore we merged these two types, resulting in the three discount types (low, medium, high).

2.2.2 Opponent Model

Implementing the Opponent Model

Using the BOA framework, the following opponent models were compared with regard to accuracy: the *Bayesian Model* [5], *Scalable Bayesian Model* [5], *HardHeaded Frequency Model* [3], and the *IAMhaggler Bayesian Model* [9]. The *IAMhaggler Bayesian Model*, which is part of the Southampton framework, was found to be the most accurate in estimating the Kalai-Smorodinsky point. Note that after we submitted our agent, we found that the *HardHeaded Frequency Model*, or a similar frequency model, would have been a better choice. Unfortunately, the performance of the model was initially measured to be poor due to a bug in GENIUS.

The computational resources required by the *IAMhaggler Bayesian Model* depend strongly on the domain size. In addition, we found that while initially learning improves the estimation, the quality of the estimation actually decreases over time on some domains after a small number of rounds. Therefore we stop updating our model after 35% of the time has passed.

Optimizing the Opponent Model

To select the best opponent model given the set of opponent models part of the BOA framework, we first implemented a set of opponent model quality measures which are used to visualize the quality of an opponent model over time. An overview of the quality measures is depicted in Table 1.

Quality measure	**Description**
Pearson correlation coefficient of bids [7]	Pearson correlation between the estimated utility of each bid for the opponent and the actual utility
Ranking distance of bids [7]	Distance between the estimated utility of each bid for the opponent and the actual utility
Absolute Kalai difference	Absolute difference between the estimated and actual utility of the Kalai-Smorodinsky point for the opponent

Table 1: Overview of quality measures used to estimate the opponent model's accuracy.

In contrast to expectation, the accuracy of the *IAMhaggler Bayesian Model* in general becomes worse over time. We believe that this can be attributed to the assumed decision function, which more accurately reflects the real decision function at the beginning of the negotiation for most agents. To improve the model's accuracy, the assumed decision function was adapted to concede less over time. This lead to better results at the beginning of the negotiation. In addition, we stop updating the model after 35% of the time. This value was found by analyzing the performance of the opponent model on a large set of domains.

2.2.3 Acceptance Strategy

Implementing the Acceptance Strategy

The acceptance strategy of TNR consists of a set of basic acceptance conditions discussed in [1]. The flowchart of the acceptance strategy is depicted in Figure 3. As visualized, there are two paths depending if the discount is negligible or not and six parameters $(\alpha, \beta, \gamma, \delta, \varepsilon, \zeta)$.

$AC_{reservation}$ is an acceptance condition which accepts when the discounted utility of the bid under consideration for offering is lower or equal to the reservation value. $AC_{constant}$ is an acceptance condition which accepts when the utility of the opponent's bid is at least equal to a constant ζ. AC_{next} accepts when a linear function of the opponent's bid utility is better than the utility of the bid under consideration. Finally, we use $AC_{max\_in\_window}$ when there is $1-\varepsilon$ time left and the utilities of the bids of the agents have not crossed. This acceptance condition compares the offered bid with the maximum bid that has been given in a particular window and will accept if is higher than the maximum given in the previous window and if it is higher than 0.5. For more detail about these acceptance conditions we refer to [1]

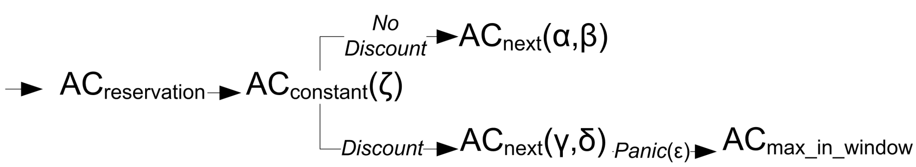

Fig. 3: Basic acceptance conditions used by TNR.

Optimizing the Acceptance Strategy

The multi-acceptance criteria (MAC) functionality of the BOA framework introduced in [2] was used to optimize the acceptance strategy. In short, the MAC can be used to run a large set of acceptance conditions in parallel during the same negotiation thread, assuming that the computational cost of each acceptance condition is minimal. Each acceptance is recorded separately, which allows us to study the performance of the different acceptance conditions.

In our test, we used the same negotiation setting used for optimizing the opponent model in Section 2.2.2. In total 288 acceptance conditions were tested varying in the usage of the panic phase and the four parameters of the two acceptance conditions AC_{next}. We found that $\alpha = 1.0, \beta = 0.0, \gamma = 1.05, \delta = 0.0, \varepsilon = 0.99$ led to be the best results.

3 Empirical Evaluation

The performance of the final agent was analyzed by running a tournament. Section 3.1 discusses the setup of this tournament and introduces the selected quality measures. Following, Section 3.1 discusses the results of the tournament in Section 3.2.

3.1 Tournament Setup

The default alternating offers protocol of GENIUS is used to run the tournament [8]. The tournament is similar to the ANAC2011 competition, except that *ValueModelAgent* is excluded and TNR is included, and that the agents compete on the 24 modified ANAC2011 domains discussed in Section 2.2.2. The full tournament is executed 10 times to increase the statistical significance of the results. In total 13440 matches were ran using the distributed version of GENIUS discussed in [2]. The computers used have similar hardware to minimize the variance of results. The set of quality measures used is shown in Table 2.

Quality measure	**Description**
Avg. time of agreement	The average time of agreement of all matches which resulted in agreement
Std. time of agreement	Standard deviation of the average time of agreement of each run
Avg. discounted utility	Average discounted utility of all matches
Std. discounted utility	Standard deviation of the average discounted utility of each run
Percentage of agreement	Percentage of matches which resulted in an agreement
Avg. Kalai distance	The average Kalai distance of all matches
Avg. unfortunate moves [6]	The average percentage of unfortunate moves of all matches
Avg. fortunate moves [6]	The average percentage of fortunate moves of all matches
Avg. nice moves [6]	The average percentage of nice moves of all matches
Avg. selfish moves [6]	The average percentage of selfish moves of all matches
Avg. concession moves [6]	The average percentage of concession moves of all matches
Avg. silent moves [6]	The average percentage of silent moves of all matches

Table 2: Overview of quality measures used to estimate the quality of a negotiation strategy.

3.2 Evaluation

This section discusses the results of the tournament visualized in Table 3. First, the table shows that for all agents the standard deviation of the time of agreement and discounted utility is negligible. Almost all matches end in agreement. TNR achieves the highest discounted utility, and strongly outperforms the runner-up. The last six measures in Table 3 are the set of trajectory measures discussed in [6]. TNR makes the least concessions, as indicated by its high percentage of silent moves and its low ranking on the percentage of unfortunate moves, fortunate moves, nice moves, and concession moves, which are all types of moves made when the agent tries to make a concession. TNR agent does not make selfish moves, which can be attributed to its usage of the time-dependent strategy.

Agent	Avg. time of agreement	Std. time of agreement	Avg. discounted utility	Std. discounted utility	Percentage of agreement	Avg. unfortunate moves	Avg. fortunate moves	Avg. nice moves	Avg. selfish moves	Avg. concession moves	Avg. silent moves
TheNegotiator Reloaded	0.545	0.001	**0.809**	**0.002**	99.911	0.033	0.000	0.033	0.000	0.003	**0.930**
Gahboninho	0.528	0.002	0.782	0.001	99.672	0.027	0.001	0.038	0.002	0.004	0.929
HardHeaded	**0.638**	0.001	0.778	0.001	99.911	0.111	0.013	**0.133**	0.052	0.028	0.663
Nice Tit For Tat Agent	0.605	0.003	0.767	**0.002**	**100.00**	0.112	0.079	0.066	0.116	0.115	0.512
Agent K2	0.493	0.002	0.755	0.001	99.821	0.154	0.116	0.069	**0.203**	0.174	0.284
The Negotiator	0.591	0.001	0.751	0.001	**100.00**	0.080	0.036	0.071	0.077	0.051	0.685
IAMhaggler2011	0.377	0.003	0.748	**0.002**	99.970	**0.162**	**0.120**	0.074	**0.203**	**0.178**	0.263
BRAMAgent	0.578	**0.004**	0.740	0.001	**100.00**	0.115	0.075	0.085	0.148	0.104	0.472

Table 3: Overview of the results of the quality measures for each agent. Bold text is used to emphasize the highest value, and underlined the lowest value. All averages are in the range $[0, 1]$.

4 Conclusion and Future Work

The tournament results discussed in Section 3 indicate a strong performance of TNR on various domains against a range of opponents. In the ANAC2012, TNR finished third overall and achieved the highest utility on undiscounted domains. The agent finished fifth when only focussing on the discounted domains. We believe that this can be attributed to our test set used to optimize the agent: ANAC2011 agents perform relatively poor on discounted domains.

In addition, as discussed in Section 2.2.2, we noted that a frequency-based opponent model such as the *HardHeaded Frequency Model* would have been a better choice to both estimate the Kalai-Smorodinsky point and select a bid for the opponent given a set of similarly preferred bids.

For future work, it could be interesting to enable TNR to identify behavior-based strategies. In this case the bidding strategy should be further extended to use an effective counter-strategy. Finally, the choice of the concession speed could be made dependent on the exact discount.

Concluding, in this work we discussed the implementation, optimization, and evaluation of a flexible negotiation strategy which outperforms the ANAC2011 agents on various domains and performs well in the ANAC2012. In addition, we presented the first ANAC agent developed using the BOA framework. For future work we plan to continue our work in designing negotiation strategies for automated negotiations.

Acknowledgments

We would like to thank Catholijn Jonker, Koen Hindriks, and Tim Baarslag for introducing us to the field of bilateral negotiation. Furthermore, we thank the Universiteitsfonds Delft and the Interactive Intelligence Group of the Delft University of Technology for sponsoring our trip to the AAMAS2012.

References

[1] T. Baarslag, K. Hindriks, and C. Jonker. Acceptance conditions in automated negotiation. *ACAN2011*, page 48, 2011.

[2] T. Baarslag, K. Hindriks, M. Hendrikx, A. Dirkzwager, and C. Jonker. Decoupling negotiating agents to explore the space of negotiation strategies. In *Proceedings of The 5th International Workshop on Agent-based Complex Automated Negotiations (ACAN 2012)*, 2012.

[3] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence Journal*, Submitted.

[4] P. Faratin, C. Sierra, and N.R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[5] K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, volume 1, pages 331–338. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[6] K. Hindriks, C. Jonker, and D. Tykhonov. Analysis of negotiation dynamics. *Cooperative Information Agents*, XI:27–35, 2007.

[7] K.V. Hindriks and D. Tykhonov. Towards a quality assessment method for learning preference profiles in negotiation. In *Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis: Aamas Workshop (2008)*, volume 44, pages 46–59. Springer-Verlag, 2010.

[8] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 2012.

[9] C.R. Williams, V. Robu, E.H. Gerding, and N.R. Jennings. Iamhaggler: A negotiation agent for complex environments. *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (Submitted)*, 2010.

Chapter 5

Evaluating the Quality of Opponent Models

One of the main directions for future work discussed in the survey of opponent models in Chapter 2, is the construction of evaluation methods to quantify the quality of existing opponent models. In line with this direction for future work, this chapter presents two papers on evaluating the quality of opponent models which estimate the opponent's preferences.

The first paper concerns a measurement method to quantify the performance gain of an opponent model relative to not using a model. The main strength of this method, is that it directly provides insight which opponent model is best in a specific setting. The paper was submitted to the 25th Australasian Joint Conference on Artificial Intelligence. We are currently waiting for an acceptance notification.

The second paper concerns a measurement method which quantifies the accuracy of an opponent model, and therefore provides better insight in how the model can be improved. Furthermore, this work discusses the relationship between the accuracy and the performance of a model. We plan to publish this paper in the near future.

Taken together, the combination of the measurement method allows us to compare the quality of existing models and decide which one is best. One of our most interesting conclusions, is that two simple types of opponent models – value models and frequency models – actually strongly outperform the Bayesian models which are popular in both literature [11] and the ANAC [1, 2].

Measuring the Performance of Online Opponent Models in Automated Bilateral Negotiation

Tim Baarslag, Mark Hendrikx, Koen Hindriks, and Catholijn Jonker

Interactive Intelligence Group, Delft University of Technology,
Mekelweg 4, Delft, The Netherlands
{T.Baarslag,K.V.Hindriks,C.M.Jonker,M.Hendrikx}@tudelft.nl

Abstract. An important aim in bilateral negotiations is to achieve a win-win solution for both parties; therefore, a critical aspect of a negotiating agent's success is its ability to take the opponent's preferences into account. Every year, new negotiation agents are introduced with better learning techniques to model the opponent. Our main goal in this work is to evaluate and compare the performance of a selection of state-of-the-art online opponent modeling techniques in negotiation, and to determine under which circumstances they are beneficial in a real-time, online negotiation setting. Towards this end, we provide an overview of the factors influencing the quality of a model, and we analyze how the performance of opponent models depends on the negotiation setting. This results in better insight into the performance of opponent models, and allows us to pinpoint well-performing opponent modeling techniques which did not receive much previous attention in literature.

Keywords: Negotiation, Opponent Model Performance, Quality Measures

1 Introduction

A negotiation between two agents is a game in which both agents try to reach an agreement better than their status quo. To avoid exploitation, agents often keep their preferences private during the negotiation [6]; however, if an agent has no knowledge about its opponent's preferences, then this can result in a suboptimal outcome [9]. A common technique to counter this is *learning* the opponent's preference profile during the negotiation, which aids in increasing the quality of the negotiation outcome by identifying bids that are more likely to be accepted by the opponent [6, 9, 20].

If there have been previous negotiations with a similar opponent, the opponent model can be prepared *before* the start of the actual negotiation; we will refer to these models as *offline* models (for example [6]). Contrastingly, if the agent has to learn the preferences *during* the negotiation it performs *online* modeling (for example [8, 9, 13]).

In this work we focus on *online* opponent models in a *single-shot* negotiation with *private* preference profiles; i.e., a setting in which an agent has no knowledge about the opponent's preference profile and no history of previous negotiations is available. There has been recent interest in opponent modeling for such settings, for example in the Automated Negotiating Agents Competition (ANAC) [1, 4]. Despite ongoing research in this area, it is not yet clear how different approaches compare, and empirical evidence has raised the question whether using an opponent model is beneficial at all in such a setting. To illustrate: state-of-the-art agents, such as the top three agents of both ANAC 2010 [4] and ANAC 2011 [1], do not model the opponent, yet outperformed agents that do. One reason that opponent modeling does not guarantee a better outcome for an agent is that the model can be a poor representation of the opponent's preferences. If the model consistently suggests unattractive bids for the opponent, it may even be preferable to not employ one at all. Secondly, a time-based deadline introduces an additional challenge for online opponent modeling, as learning the model can be computationally expensive and can therefore influence the amount of bids that can be explored. More precisely, the gain in using the model should be higher than the loss in utility due to decreased exploration of the outcome space. We will refer to this as the *time/exploration trade-off*.

Apart from the inherent trade-off in opponent modeling, we are interested whether opponent models are accurate enough to provide gains at all, even when ignoring computational costs. To this end, we evaluate opponent models in two settings: a time-based and round-based negotiation protocol. This paper compares a large set of opponent modeling techniques, which were isolated from state-of-the-art negotiation strategies. We measure their performance in various negotiation settings, and we provide a detailed overview of how the different factors influence the final negotiation outcome.

After discussing related work in Section 2, we introduce the negotiation setting and consider the difficulties in evaluating opponent models in Section 3. In Section 4 we introduce a method to quantify opponent model performance, after which we apply it to a set of models in Section 5. We formulate hypotheses and analyze the results in Section 6; and finally, in Section 7 we provide directions for future work.

2 Related Work

Opponent modeling has received a lot of attention in automated negotiation. There are three groups of related work when considering opponent model evaluation. The first category consists of work that details an agent strategy in which the opponent model is introduced, but the performance is not evaluated. Examples of this type are [8] and [19].

The second category compares a single novel model to a set of baseline strategies. The approaches usually differ in how they define performance. In [9] for example, a model is introduced for the same time-based protocol discussed in this work. The performance of the opponent model is estimated by embedding it in a strategy and comparing the average utility against two baseline strategies. The modeling technique discussed by [15] introduces a model for a similar protocol, but in this case the baseline is set by humans. Zeng and Sycara measure performance in terms of social welfare, but focus on single-issue negotiations in which they compare the performance of three settings: both learn, neither learn, and only the buyer learns [20]. Finally, [5] evaluates the accuracy of a model against simple baseline strategies in terms of the likelihood that the correct class is estimated to which the opponent's preference profile belongs.

The last category is most similar to our work, and consists of literature comparing the performance of a model against other models or against a theoretical lower or upper bound. For example, Coehoorn and Jennings [6] evaluate the performance of their opponent model using a standard bidding strategy which can be used both with and without a model. The performance of the strategy is evaluated in three settings: without knowledge, with perfect knowledge, and when using an offline opponent model. This work is similar to our work, however, it differs in the fact that we focus on *online* opponent modeling. Our setting is especially challenging as it involves the time/exploration trade-off. Another example is the work by [12], which introduces two opponent models for e-recommendation in a multi-object negotiation. Compared to our work, we focus on the more general type of multi-issue negotiations. Finally, [10] defines two accuracy measures and uses these measures to analyze the accuracy of two opponent models. The main differences are that we focus on a larger set of performance measures, and pay more attention to the factors that influence the performance of the model.

Furthermore, as far as we know, our work is the first to compare and analyze such a large set of state-of-the art models of the opponent's preference profile.

3 Evaluating Opponent Models

The main goal of this work is to answer the question: "*Under what circumstances is it beneficial to use an online opponent model in a real-time negotiation setting?*". An answer is not straightforward due to the time/exploration trade-off and potentially poor accuracy of a model. In particular, we want to answer the following research questions:

1. Assuming *perfect* knowledge about the opponent's preferences, is there a significant performance gain in using this information compared to ignoring it?
2. Is there a significant performance gain from using an *online opponent model* in comparison to *not using a model*, assuming no prior knowledge is available?

The main difficulty in finding a conclusive answer to these questions, is that the performance of an opponent model depends on the negotiation setting. Therefore, we study an third, overarching research question:

3. How does the performance of using an opponent model depend on the setting?

3.1 Preliminaries

In this work we focus on a bilateral automated negotiation in which two agents try to reach an agreement while maximizing their own utility. Agents use the widely-employed alternating-offers protocol for bilateral negotiations [16], in which the negotiating parties take turns in exchanging offers. A negotiation scenario consists of the negotiation domain, which specifies the setting and all possible bids, together with a privately-known preference profile for each party. A preference profile is described by a utility function $u(x)$, which maps each possible outcome x in the negotiation domain to a utility in the range $[0, 1]$. In this work we discuss opponent models that attempt to estimate the opponent's utility function $u'(x)$ during the negotiation.

3.2 Influence of the Agent's Strategy

Different agents apply their opponent model in different ways. There are two main factors in which the application of an opponent model by a bidding strategy can differ:

– *Type of information gained from the opponent model*. A bidding strategy can employ an opponent model for different reasons: for example, it can be employed to select the best bid for the opponent out of a set of similarly preferred bids [3, 19]; or to select a bid that optimizes a weighted combination of both utility functions [8]; or it can help to estimate the utility of a specific outcome such as the Nash-point [3].
– *Selecting a bid using an opponent model*. When a model is used to select a bid from a set of similarly preferred bids, the question still remains what selection criteria to use. One straightforward solution is to select the *best* bid for the opponent, but this may not be optimal, as in general opponent models are imperfect. An alternative is to select a random bid from the set of n best bids [3].

Even when the factors above are taken into account, still care has to be taken to properly compare different models. Opponent models can only be fairly compared if the other components, such as bidding strategy and acceptance strategy [2] are fixed.

3.3 Influence of the Opponent's Strategy

All opponent modeling techniques make certain assumptions about the opponent, so as to assign meaning to the observed behavior. If the opponent does not adhere to these assumptions, the model may not reflect reality well. The set of strategies against which a model is tested is a decisive factor when measuring its performance. Therefore, a set of opponents should contain both agents that fulfill the model's assumptions to determine its efficacy in optimal conditions; and agents that test the model's robustness by violating its assumptions.

The following assumptions were found by analyzing the models in Section 5.2:

1. *The concession of the opponent follows a particular function*. Some opponent modeling techniques assume that the opponent uses a given time-based bidding strategy. Modeling the opponent then reduces to estimating all issue weights such that the predicted utility by the modeled preference profile is close to the assumed utility.
2. *The first bid made by the opponent is the most preferred bid*. The best bid is the selection of the most preferred value for each issue, and thereby immediately reveals which values are the best for each issue. Many agents start with the best bid.
3. *There is a direct relation between the preference of an issue and the times its value is significantly changed*. To learn the issue weights, some models assume that the amount of times the value of an issue is changed is an indicator for the importance of the issue. The validity of this assumption depends on the distribution of the issue and value weights of the opponent's preference profile and its bidding strategy.
4. *There is a direct relation between the preference of a value and the frequency it is offered*. A common assumption to learn the value weights is to assume that values which are more preferred are offered more often. Similar to the issue weights assumption, this assumption strongly depends on the agent's strategy and domain.

3.4 Influence of the Negotiation Scenario

Three main factors of a scenario influence the quality of an opponent model:

1. *Domain size*. In general, the larger the domain, the less likely a bid is a Pareto-bid. Furthermore, domains with more bids are likely more computationally expensive to model. Therefore, the influence of the time/exploration trade-off is higher.
2. *Bid distribution*. The bid distribution quantifies how bids are distributed. We define bid distribution as the average distance of all bids to the nearest Pareto-bid. The bid distribution directly influences the performance gain attainable by a model.
3. *Opposition*. We define opposition as the distance from the Kalai-point to complete satisfaction $(1, 1)$. The opposition of a domain influences the number of possible agreements, and opponent models may be help in locating them more easily.

4 Measuring the Performance of Opponent Models

As we noted in the previous section, the effectiveness of an agent's opponent model is heavily influenced by the negotiation setting. This work proposes a careful measurement method of opponent modeling performance, and can be interpreted as a first step towards creating a generic performance benchmark for the type of opponent models that we study here. The following sections discuss the four components of the method.

4.1 Negotiation Strategies of the Agents

For the negotiation strategies of the agents in which the opponent models are embedded, we elected a variant of the standard time-dependent tactic [7]. This strategy is chosen for its simple behavior, which elicits regular behavior from its opponents; furthermore, adding a model may significantly increase its performance. Given a target utility, the adapted agent generates a set of similarly preferred bids and then selects a bid using the opponent model. We focus on selecting a bid from a set of similarly preferred bids, as this usage is commonly applied, for example in [19] and [13]. We embedded the models in four time-dependent agents ($e = 0.1; 0.2; 1.0; 2.0$). We opted for multiple agents as we believed that the concession speed can influence the performance gain.

The remaining issue in using an opponent model is which bid to select for the opponent given a set of similarly preferred bids. Given the approaches in Section 3.2, we opted to have the models select the best bid for the opponent, as this approach is most differentiating: it leads to better performance of the more accurate opponent models.

4.2 Negotiation Strategies of the Opponents

This section discusses the opponents selected using the guidelines outlined in Section 3.3. The set of opponent strategies consists of three cooperative agents, which should be easy to model as their concession speed is high, and five competitive agents. The set of conceding agents consists of two *time-dependent agents* with high concession speeds $e \in \{1, 2\}$, and the *offer decreasing* agent, which offers the set of all possible bids in decreasing order of utility. The set of competitive agents contains two *time-dependent agents* with low concession speeds $e \in \{0.0, 0.2\}$, and the ANAC agents *Gahboninho*, *HardHeaded*, and *IAMcrazyHaggler*.

Given the five opponent modeling assumptions introduced in Section 3.3, the first assumption about the opponent's decision function fails in general, as an opponent in practice never completely adheres to the assumed decision function. The second assumption holds for all agents except *IAMcrazyHaggler*, whose first bid is randomly picked. The other three assumptions are typical for the frequency models. It is not possible to adhere to or violate these assumptions completely, as they depend both on the negotiation scenario structure and opponents.

4.3 Negotiation Scenarios

As we explored in Section 3.4, the *domain size*, *bid distribution*, and *opposition* of a negotiation scenario are all expected to influence an opponent model's performance, and therefore we aimed for a large spread of the characteristics of the scenarios, as visualized in Table 1. In total seven negotiation scenarios were selected from ANAC and existing literature.

Scenario name	**Size**		**Bid distrib.**		**Opposition**	
ADG [1]	15625	(*med.*)	0.136	(*low*)	0.095	(*low*)
Grocery [1]	1600	(*med.*)	0.492	(*high*)	0.191	(*med.*)
IS BT Acquisition [1]	384	(*low*)	0.121	(*low*)	0.125	(*low*)
Itex–Cypress [11]	180	(*low*)	0.222	(*med.*)	0.431	(*high*)
Laptop [1]	27	(*low*)	0.295	(*med.*)	0.178	(*med.*)
Employment contract [18]	3125	(*med.*)	0.267	(*med.*)	0.325	(*high*)
Travel [4]	188160	(*high*)	0.416	(*high*)	0.230	(*med.*)

Table 1. Characteristics of the negotiation scenarios.

4.4 Quality Measures for Opponent Models

The quality of an opponent model can be measured in two ways: a black box approach, in which *performance measures* evaluate the ultimate goal, namely the quality of the outcome; and there is the white box view, which uses *accuracy measures* capable of considering the internal design of a strategy and revealing the accuracy of the estimation of the opponent's preference profile.

This work focuses on performance measures, as [10] has already compared opponent models using a white box approach, albeit in a more limited setting. Furthermore, more accurate models do not necessarily lead to improved performance; however, we also discuss accuracy measures in future work.

In this work we employ six performance measures. First, the *utility performance* of an individual opponent model, which is measured as the average score of the agents employing it against the selected opponents on all negotiation scenarios. We also measure the *average time of agreement* and the *average amount of rounds* that the negotiation takes. Finally, we test the *average distance from the outcome to the Pareto-frontier, Kalai-point, and Nash-point* of all negotiations that result in an agreement.

5 Experiments

We applied the method described in the previous section to our experimental setup below in order to answer the research questions introduced in Section 3.

5.1 Experimental Setup

To analyze the performance of different opponent models, we employed GENIUS [14], which is an environment that facilitates the design and evaluation of automated negotiators' strategies and their components. The experiments are subdivided into two categories: we use a standard *time-based protocol*, as well as a *round-based protocol*. In total, we ran 17920 matches, which on a single computer takes nearly two months.

Our main interest goes out to the real-time setting, as this protocol features the time/exploration trade-off. We applied our benchmark to the set of models using the time-based protocol. Each match features a real-time deadline set at three minutes.

In the round-based protocol the same approach is applied, but in this case, time does not pass within a round, giving the agent infinite time to update its model. This provides valuable insights into the best *theoretical* result an opponent model can achieve.

5.2 Opponent Models

We compare the performance of the opponent models used in ANAC [1, 4], which is a yearly international competition in which negotiating agents compete on multiple domains. Each year, the competition leads to the introduction of new negotiation strategies with novel opponent models. While the domain (i.e. the set of outcomes) is common knowledge to all agents, the utility function of each player is private information and hence has to be learned. The utility functions of the agents are *linearly additive*; that is, the overall utility consists of a weighted sum of the utility for each individual issue. The setting of ANAC is consistent with the preliminaries in this paper.

We specifically opted to use agents that participated in ANAC for the following reasons: the agents are designed for one consistent negotiation setting which makes it possible to compare them fairly; their implementation is publicly available; and finally, we believe that the agents and opponent models represent the current state-of-the-art. We used modeling techniques from ANAC 2010 [4], ANAC 2011 [1], and a selection of opponent models designed for ANAC 2012. We isolated the opponent models from the agents and reimplemented them as separate generic components to be compatible with all other agents (as in [2]). As discussed in Section 3.2, this setup allows us to equip a single negotiation strategy with various opponent models, which makes it straightforward to fairly compare the different modeling techniques.

Table 2 provides a summary of the opponent models used in our experiments. We did not include the *Bayesian Model* from [9] and the *FSEGA Bayesian Model* [17], even though they fitted our setup, as both models were not designed to handle domains containing more than a 1000 bids. We are aware that many alternative opponent modeling techniques exist [5, 9, 15, 20]; however, for our negotiation setting, this is currently the largest set available of comparable opponent modeling techniques.

Based on our analysis, we found that in our selection two approaches to opponent modeling are prominent: *Bayesian opponent models* and *Frequency models*.

Bayesian opponent models generate hypotheses about the opponent's preferences [9]. The models presuppose that the opponent's strategy adheres to a specific decision function; for example a time-dependent strategy with a linear concession speed. This is then used to update the hypotheses using Bayesian learning.

Frequency models learn the issue and value weights separately. The issue weights are usually calculated based on the frequency that an issue *changes* between two offers. The value weights are oten calculated based on the frequency of *appearance* in offers.

Both modeling approaches are prone to failure as they rely on a subset of the assumptions introduced in Section 3.3. More specifically, Bayesian models make strong assumptions about the opponent's strategy, whereas frequency models assume knowledge about the value distribution of the issues of a preference profile and place weak restrictions on the opponent's negotiation strategy. Generally, the Bayesian models are far more computationally expensive; however, it is unknown if they are more accurate.

Model	**Description**	**M**
No Model	No knowledge about the preference profile.	-
Perfect Model	Perfect knowledge about the preference profile.	-
Bayesian Scalable Model [9]	This model learns the issue and value weights separately. The opponent is assumed to concede a constant amount per turn.	1
IAMhaggler Bay. Model [19]	This model uses an efficient Bayesian learning technique in which the opponent is assumed to use a specific time-dependent decision function.	1
HardHeaded Freq. Model [13]	This model learns the issue weights based on how often the values change. The value weights are learned based on frequency.	3 4
Smith Freq. Model [8]	Uses a similar approach to the *HardHeaded Frequency Model*, but far less computationally efficient.	3 5
Agent X Freq. Model	This model is a more complex variant of the *HardHeaded Frequency Model*, which also takes the opponent's repeated bids into account.	3 4
N.A.S.H. Freq. Model	In contrast to *HardHeaded Frequency Model*, this model learns the issue weights based on the frequency that the assumed best value is offered.	2 4

Table 2. Overview of the opponent models and their modeling assumptions (M).

6 Results

Below we analyze the outcomes of the experiment to provide an answer to the research questions in the form of hypotheses **H1–H6**. We first discuss the overall gain in performance when using perfect knowledge versus online opponent modeling. Section 6.2 provides an answer to the final research question on how the negotiation setting influences the performance of an opponent model.

6.1 Overall Performance of Opponent Models

Our experimental results for a selection of the quality measures described in Section 4.4 are shown in Table 3 for both the time-based and round-based protocol. Before we analyze the performance gain of online opponent models, we first answer the question whether perfect knowledge aids in improving the negotiation outcome at all:

H1. *Usage of the perfect model by a negotiation strategy leads to a significant performance gain in comparison to not using an opponent model.*

We expected that perfect knowledge about the opponent's preference profile would significantly improve performance of an agent. Our main aim here was not to reconfirm the already widely acknowledged benefits of integrative bargaining, but to analyze whether our experimental setup is a valid instrument for measuring the learning effect in other types of settings. Our expectation is confirmed by the experiment, as the perfect model yields a significant performance increase on all quality measures (except average rounds) for both negotiation protocols. For the real-time protocol, the difference between the best online opponent (*HardHeaded Frequency Model*) and *No Model* is 0.0135; for the round-based protocol it is 0.0144 (*Smith Frequency Model*). Note that while the gains are small, there are three small domains where opponent modeling does not result in significant gains. If we solely focus on the large *Travel* negotiation scenario, then the gain relative to *No Model* becomes 0.0413 for the *Perfect Model*. Especially note the improvement in distance between the outcome and Pareto-frontier, and the earlier agreements, in Table 3. This leads us to conclude that using an opponent model leads to better performance as it aids in increasing the quality of the outcome.

H2. *Usage of an online opponent model leads to a significant performance gain when time is not an issue. Online opponent modeling does not yield the same benefit in a real-time setting because of the time/exploration trade-off.*

We noted previously that in some cases, ANAC agents that do not model the opponent can outperform agents that do, and such agents have even won the competition. This led us to believe that online modeling does not benefit the agents, either because it misrepresents the preferences, or by taking too much time in a time-sensitive setting.

This is why it came as a surprise that in *both* the time- and round-based protocol, online opponent models performed significantly better on all quality measures. For the time-based protocol the best online opponent models are the frequency models, except for the *Smith Frequency Model* who scores badly in this case. However, for the round-based protocol, the *Smith Frequency Model* is actually best. This is caused by the time/exploration trade-off, because the model is computationally expensive as indicated by the small amount of bids offered in the time-based protocol.

Surprisingly the worst performance on a quality measure is not always made by using *No Model*. For example in the time-based experiment the *Bayesian Scalable Model* has the worst performance. The Bayesian model of *IAMhaggler* however, performs much better, but disappoints in the round-based protocol. We believe this can be attributed to its updating mechanism: only unique bids are used to update the model,

Quality Measures	Perfect	HH. FM	Agent X FM	Nash FM	IAH. BM	Smith FM	None	Scal. BM
Time-based								
Avg. utility	<u>.7285</u>	.7260	.7257	.7257	.7178	.7156	.7125	**.7077**
Avg. time of agr.	**.4834**	.4865	.4867	.4865	.4958	.4937	.5022	<u>.5055</u>
Avg. rounds	7220	7218	7231	7198	7004	**4745**	<u>7352</u>	4836
Avg. Pareto dist. of agr.	**.0007**	.0017	.0015	.0018	.0069	.0068	.0059	<u>.0071</u>
Avg. Kalai dist. of agr.	**.2408**	.2434	.2447	.2428	.2515	.2474	<u>.2683</u>	.2561
Avg. Nash dist. of agr.	**.2442**	.2471	.2481	.2483	.2541	.2500	<u>.2721</u>	.2594
Rounds-based								
Avg. utility	<u>.7235</u>	.7196	.7191	.7192	.7111	.7199	**.7050**	.7124
Avg. time of agr.	**.4928**	.4975	.4978	.4977	.5058	.4974	<u>.5136</u>	.5038
Avg. rounds	**2508**	2531	2533	2533	<u>2572</u>	2531	2567	2562
Avg. Pareto dist. of agr.	**.0010**	.0029	.0023	.0028	<u>.0073</u>	.0026	.0066	.0063
Avg. Kalai dist. of agr.	**.2332**	.2380	.2395	.2380	.2456	.2369	<u>.2614</u>	.2445
Avg. Nash dist. of agr.	**.2370**	.2403	.2437	.2404	.2516	.2403	<u>.2644</u>	.2472

Table 3. Performance of all models on a set of quality measures for both protocols.

which speeds-up updating but can result in poor performance against slowly conceding agents which offer the same bid multiple times.

In conclusion, online opponent model can result in significant gains and surprisingly, frequency models lead to the largest gains, outperforming the Bayesian models. We believe that the winners of ANAC could have performed even better by learning the opponent's preferences with a frequency model. The success of the frequency model can be attributed to its simplicity and hence faster performance, and to the fact that it is more robust by making weaker assumptions about the strategy of the opponent in comparison to the Bayesian modeling approaches.

6.2 Influence of the Negotiation Setting

We will now discuss the influence of each of the three components of the negotiation setting on the quality of an opponent model, following the structure of Section 3.

Influence of the Agent's Strategy. The performance gain of using an opponent model necessarily depends on the strategy in which it is embedded. Table 4 provides an overview of the relative gain in comparison to *No Model* for all opponent models in the time-based experiment. Based on the results, we have tested the following hypothesis:

H3. *The more competitive an agent, the more it benefits from using an opponent model.*

At each turn of a negotiation session, a set of possible agreements can be defined. This is the intersection of two sets: the set of bids which an agent considers for offering, and the set of all bids acceptable to the opponent. The more competitive the agent, the smaller the intersection between the two sets. When an agent concedes, the number of possible agreements increases at the cost of utility. An opponent model can help in finding possible agreements, preventing concession and therefore loss in utility. We therefore expected the gain for competitive agents to be higher, as the set of possible agreements each turn is smaller, and therefore an optimal bid is more easily missed by an agent not employing an opponent model. This is especially decisive in the last few seconds of the negotiation, when many agents concede rapidly to avoid non-agreement.

The hypothesis is confirmed by our experiments. In Table 4 there is a negative correlation between the concession speed and relative gain in performance. If we ignore the results of the three worst performing models, a small – albeit statistically significant – negative correlation of -0.508 is found between the concession speed and the performance gain of an agent.

Agents	$\mathbf{e = 0.1}$	$\mathbf{e = 0.2}$	$\mathbf{e = 1}$	$\mathbf{e = 2}$
Perfect Model	0.0180	0.0164	0.0152	0.0144
HardHeaded Freq. Model	0.0156	0.0137	0.0118	0.0128
Agent X Freq. Model	0.0161	0.0137	0.0116	0.0113
N.A.S.H. Freq. Model	0.0166	0.0129	0.0108	0.0121
IAMhaggler Bay. Model	0.0084	0.0055	0.0033	0.0039
Smith Freq. Model	-0.0031	0.0020	0.0071	0.0063
Bayesian Scalable Model	**-0.0050**	**-0.0058**	**-0.0032**	**-0.0053**

Table 4. Utility of each opponent model relative to using *No Model* for each agent.

Influence of the Opponent's Strategy. The opponent's behavior also has an important impact on the performance of an opponent model. Based on the results shown in Table 5, we test the three hypotheses below.

H4. *An agent benefits more from an opponent model against competitive agents.*

Intuitively, the more competitive the opponent, the more useful the opponent model as the set of possible agreements is smaller, analogous to hypothesis **H3**. Therefore, we expected the highest gain against the competitive agents *Gahboninho V3*, *HardHeaded*, and *IAMcrazyHaggler*. However, in Table 5 only the gain for *Gahboninho V3* and *IAMcrazyHaggler* is very high.

For *HardHeaded*, we believe this can be attributed to the agent using an opponent model itself. If the opponent uses a well-performing opponent model, then the performance gain of an opponent model can be expected to be lower, as the opponent is already able to make Pareto-optimal bids. Our experiment appears to the confirm this hypothesis in the case of playing against *HardHeaded*, whose well-performing opponent model seems to diminish the effect of opponent modeling by the other side.

Concluding, given the results of our experiment, we believe that the hypothesis holds, at least for consistently competitive opponents without an opponent model.

H5. *Frequency models are more robust against opponents employing a random tactic than the Bayesian models.*

Opponents	**TDT 0**	**TDT 0.2**	**TDT 1.0**	**TDT 2.0**	**OD**	**Gah.**	**HH.**	**IcH.**
Perfect	.0085	.0015	.0008	.0022	.0060	.0676	.0015	.0399
HH. Freq. Model	.0085	.0013	-.0002	.0019	.0060	.0515	.0000	.0388
Agent X Freq. Model	.0085	.0019	.0002	.0036	.0058	.0561	.0009	.0285
N.A.S.H. Freq. Model	.0085	.0005	-.0005	.0020	.0065	.0507	.0037	.0336
IAH. Bay. Model	**.0000**	.0003	-.0021	-.0001	-.0046	.0511	.0039	**-.0066**
Smith Frequency	-.0038	-.0023	-.0019	.0007	-.0113	**.0357**	**-.0224**	.0297
Bay. Scalable Model	**.0000**	**-.0033**	**-.0055**	**-.0058**	**-.0535**	.0458	-.0128	-.0036

Table 5. Utility of each opponent model relative to using *No Model* for each opponent.

In order to estimate the opponent's utility of a certain bid, both types of models make certain assumptions about the opponent. The Bayesian opponent models assume that the opponent follows a particular decision function through time (cf. modeling assumption 1 in Section 3.3), while the frequency models assume higher valued bids are offered more often (cf. modeling assumptions 3 and 4). Many opponent strategies do not adhere to these assumptions, which causes the learning models to make wrong predictions when playing against them. For example, opponents such as *IAMcrazyHaggler* who employ a random negotiation strategy, explicitly violate the assumptions of both models. For the Bayesian learning models, this means the opponent preferences will be estimated incorrectly, and more so through time. The frequency models however, are much more robust, not only in the sense that a negotiation tactic has a greater chance to satisfy its assumptions, but more significantly: it is less sensitive to a tactic violating its assumptions. For instance, in the case of *IAMcrazyHaggler*, it will deduce that it equally prefers any bid it has offered so far – which, in this case, is exactly right.

We therefore expected relatively poor performance from the Bayesian models. This hypothesis is confirmed by our experiment: the frequency models have a high performance gain against *IAMcrazyHaggler*, whereas using the Bayesian models is even worse than not using an opponent model at all.

Influence of the Negotiation Scenario. The performance of an opponent model is influenced by the characteristics of the negotiation scenario, such as amount of bids, distribution of the bids, and the opposition of the domain. Table 6 provides an overview of the relative gain of all opponent models in comparison to *No Model* for in the time-based experiment. Based on these results, we formulate the following hypothesis:

H6. *The higher the amount of bids, bid distribution, or opposition of a scenario, the more an agent benefits from using an opponent model.*

We anticipated the bid distribution to be the major factor determining the performance gain of an opponent model. If the bid distribution is high, then the Pareto-frontier is more sparse. This means a higher gain can be expected of utilizing an opponent model to locate bids close to the Pareto-frontier. This hypothesis is confirmed by our experiments, as we found a strong Pearson correlation of 0.778 between the bid distribution

	Model	Low	Medium	High
Size	*Perfect*	0.001	0.022	0.041
	Best 4	0.002	0.018	0.039
Bid Distribution	*Perfect*	0.001	0.013	0.035
	Best 4	-0.001	0.010	0.034
Opposition	*Perfect*	0.001	0.023	0.020
	Best 4	-0.001	0.022	0.016

Table 6. Gain of each model relative to using *No Model* for each scenario parameter.

and the performance gain of the best four models, and 0.701 if we solely focus on the perfect opponent model. Therefore we confirm this sub-hypothesis.

Another factor is the size of the negotiation domain. If a domain contains more bids, then there are relatively less bids that are Pareto-optimal, so an opponent model can aid more in identifying them. On the other hand, opponent models are more computationally expensive on the larger domains. Despite this effect, we found a strong Pearson correlation between the amount of bids and the performance gain: 0.631 for the best four models, and 0.596 when using the perfect model.

The final factor is the opposition of the scenario. Intuitively, if the opposition is higher, then there are less possible agreements. Opponent models can aid in identifying these rare acceptable bids, thereby preventing break-offs, or unnecessary concessions. Nevertheless, if the opposition is high, then the bids are also relatively closer to the Pareto-optimal frontier, which renders it more difficult for an opponent model to make a significant impact on the negotiation outcome. Despite this effect, we expected that higher opposition would lead to higher performance gain. However, in our experiments we noted only a small positive Pearson correlation of 0.256 for the best four models and 0.262 for the perfect model. Based on these results we are unable to draw a conclusion, which leads us to believe the two mentioned effects cancel each other out, making the other two characteristics of the scenario decisive in the effectiveness of a model.

7 Conclusion and Future Work

This paper evaluates and compares the performance of a selection of state-of-the-art online opponent models. The main goal of this work is to evaluate if, and under which circumstances, opponent modeling is beneficial.

Measuring the performance of an opponent model is not trivial, as the details of the negotiation setting affects the effectiveness of the model. Furthermore, while we know an opponent model improves the negotiation outcome in general, the role of time should be taken into account when considering *online* opponent modeling in a real-time negotiation because of the time/exploration trade-off: a computationally expensive model may produce predictions of better quality, but in a real-time setting it may lead to less bids being explored, which may harm the outcome of the negotiation.

Based on an analysis of the contributing factors to the quality of an opponent model, we formulated a measurement method to quantify the performance of online opponent models and applied it to a large set of state-of-the-art opponent models. We analyzed two main types of opponent models: frequency models and Bayesian models. We noted that the time/exploration trade-off is indeed an important factor to consider in opponent model design of both types. However, we found that the best performing models did not suffer from the trade-off, and that most – but not all – online opponent models result in a significant improvement in performance compared to not using a model; not only because the deals are made faster, but also because the outcomes are on average significantly closer to the Pareto-frontier. A main conclusion of our work is that we noted that frequency models consistently outperform Bayesian models. This is not only because they are faster, because the effect remains in a round-based setting. This suggests that frequency models combine the best of both worlds. Surprisingly, despite their performance, frequency models have not received much attention in literature.

Our other main conclusion concerns the effects of the negotiation setting on an opponent model's effectiveness. We found that the more competitive an agent, or its opponent, the more benefit an opponent model provides. In addition, we found that the higher the size or the bid distribution of a scenario, the higher the gain of using a model.

For future work, it would be interesting to examine other uses of opponent modeling, such as opponent prediction. Another direction of future work is to investigate the interaction between opponent model performance and its accuracy through time. We also plan to test a larger set of models derived from literature and ANAC 2012.

Acknowledgements

This research is supported by the Dutch Technology Foundation STW, applied science division of NWO and the Technology Program of the Ministry of Economic Affairs. It is part of the Pocket Negotiator project with grant number VICI-project 08075.

References

1. Baarslag, T., Fujita, K., Gerding, E.H., Hindriks, K., Ito, T., Jennings, N.R., Jonker, C., Kraus, S., Lin, R., Robu, V., Williams, C.R.: Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. Artificial Intelligence Journal (Accepted)
2. Baarslag, T., Hindriks, K., Hendrikx, M., Dirkzwager, A., Jonker, C.: Decoupling negotiating agents to explore the space of negotiation strategies. In: Proceedings of The 5th International Workshop on Agent-based Complex Automated Negotiations (ACAN 2012) (2012)
3. Baarslag, T., Hindriks, K., Jonker, C.M.: A tit for tat negotiation strategy for real-time bilateral negotiations. In: New Frontiers for Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence. Springer-Verlag (In Press)
4. Baarslag, T., Hindriks, K., Jonker, C.M., Kraus, S., Lin, R.: The first automated negotiating agents competition (ANAC 2010). In: New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence. Springer-Verlag (2012)
5. Buffett, S., Spencer, B.: A bayesian classifier for learning opponents' preferences in multi-object automated negotiation. Electronic Commerce Research and Applications **6**(3), 274–284 (2007)
6. Coehoorn, R., Jennings, N.: Learning an opponent's preferences to make effective multi-issue negotiation trade-offs. In: Proceedings of the 6th international conference on Electronic commerce, pp. 59–68. ACM (2004)

7. Faratin, P., Sierra, C., Jennings, N.: Negotiation decision functions for autonomous agents. Robotics and Autonomous Systems **24**(3-4), 159–182 (1998)
8. van Galen Last, N.: Agent smith: Opponent model estimation in bilateral multi-issue negotiation. New Trends in Agent-Based Complex Automated Negotiations pp. 167–174 (2012)
9. Hindriks, K., Tykhonov, D.: Opponent modelling in automated multi-issue negotiation using bayesian learning. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (2008)
10. Hindriks, K., Tykhonov, D.: Towards a quality assessment method for learning preference profiles in negotiation. In: Agent-Mediated Electronic Commerce and Trading Agent Design and Analysis: Aamas Workshop (2008), vol. 44, pp. 46–59. Springer-Verlag (2010)
11. Kersten, G.E., Zhang, G.: Mining inspire data for the determinants of successful internet negotiations. Central European Journal of Operational Research (2003)
12. Klos, T., Somefun, K., La Poutré, H.: Automated interactive sales processes. IEEE Intelligent Systems **26**, 54–61 (2010)
13. van Krimpen, T., Looije, D., Hajizadeh, S.: Hardheaded. In: New Frontiers for Agent-based Complex Automated Negotiation, Series of Studies in Computational Intelligence (to appear)
14. Lin, R., Kraus, S., Baarslag, T., Tykhonov, D., Hindriks, K., Jonker, C.M.: Genius: An integrated environment for supporting the design of generic automated negotiators. Computational Intelligence (2012)
15. Lin, R., Kraus, S., Wilkenfeld, J., Barry, J.: Negotiating with bounded rational agents in environments with incomplete information using an automated agent. Artificial Intelligence **172**(6-7), 823–851 (2008)
16. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica: Journal of the Econometric Society **50**, 97–109 (1982)
17. Serban, L.D., Silaghi, G.C., Litan, C.M.: Agentfsega - time constrained reasoning model for bilateral multi-issue negotiations. In: New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (to appear) (2010)
18. Thompson, L.: The Mind and heart of the negotiator, 3 edn. Prentice Hall Press, Upper Saddle River, NJ, USA (2000)
19. Williams, C., Robu, V., Gerding, E., Jennings, N.: Iamhaggler: A negotiation agent for complex environments. New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence (Submitted) (2010)
20. Zeng, D., Sycara, K.: Bayesian learning in negotiation. International Journal of Human-Computers Studies **48**(1), 125–141 (1998)

Measuring and Modeling what Matters in Automated Negotiations

Tim Baarslag
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
T.Baarslag@tudelft.nl

Mark Hendrikx
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
M.Hendrikx@ii.tudelft.nl

Koen Hindriks
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
K.V.Hindriks@tudelft.nl

Catholijn Jonker
Interactive Intelligence Group
Delft University of Technology
Mekelweg 4, Delft, The Netherlands
C.M.Jonker@tudelft.nl

ABSTRACT

Opponent models are an important component in the development of efficient negotiation strategies, as they aid in increasing the quality of the negotiation outcome. A large set of opponent models have already been introduced, but to date, opponent models are mainly evaluated as black-box components that aim to improve the *performance* of a negotiation strategy. Such an approach does not fully provide insight into the reason why certain opponent models work better than others. In contrast, in this work we introduce and apply a method to quantify the *accuracy* of an opponent model. Our approach is novel in the sense that we quantify the accuracy of a set of opponent models in various ways, and we analyze how the accuracy changes over time depending on the negotiation setting. Furthermore, we determine in what way improved accuracy translates into increased performance, and we provide a set of measures that we believe are most useful to include in a benchmark.

As we plan to publish this paper in a conference this year, the version of the paper in this thesis is limited to an abstract. The committee members graded the full paper. A full version of the paper can be requested by contacting the authors.

Chapter 6

Conclusion and Future Work

There is an increasing interest in the research field of automated negotiation, mainly driven by cost reduction due to automation [5, 15, 21]. One of the main subjects of automated negotiation is the design of effective negotiation strategies. Opponent models are an important part of effective negotiation strategies, as they can aid in taking the opponent into account.

Many types of opponent models have been introduced in literature, however, up till now there was no recent survey providing structure and direction to the field. Towards this end, in Chapter 2 we identified and discussed six types of opponent models. We provided directions for future work for each type of opponent model. An important direction for future work is the design of benchmarks to evaluate and compare models of the same type.

In line with this direction, the goal of this thesis was to improve the state of the art by creating a method to quantify the quality of opponent models which estimate the opponent's preferences. An important aspect required to estimate the quality of an model, is the possibility to switch the opponent model of an existing strategy for another or no model. Towards this end, we introduced the BOA framework in Chapter 3.

The BOA framework is a framework that distinguishes the bidding strategy, the opponent model, and the acceptance strategy in automated negotiation strategies and allows to recombine these components to systematically explore the space of automated negotiation strategies. For this framework we decomposed the strategy of a large set of agents in the ANAC [1, 2]. In total the components of more than twenty agents were isolated and adapted to be compatible with framework. An important direction for future work is to extend the BOA framework to include a component which estimates the opponent's bidding strategy.

In Chapter 4 we discussed how the BOA framework was applied to create a negotiation agent which participated in the ANAC 2012. The agent finished third in the competition, which illustrates that the framework can be used to create state of the art negotiation agents.

Furthermore, based on the BOA framework, in chapter 5 we introduced a method to quantify the quality of an opponent in terms of performance gain relative to not using an opponent model. The method is based on an analysis of how the performance of an opponent model depends on in which agent it is embedded, the characteristics of the opponent, and the parameters of the negotiation scenario. We applied the method to a large set of opponent models derived from the ANAC. Surprisingly, some of the models we evaluated

– which received much attention in literature – had a negative utility gain, whereas others – which have not received attention – were not far from theoretically optimal.

However, while we found the performance gain of a large set of models, the question remained how accurate the models are and how their accuracy changes during a negotiation. Therefore, we introduced an additional method to quantify the accuracy of opponent models over time using a large set of accuracy measures. Surprisingly, we found that while in literature it is believed that opponent models improve due to learning, we found that against a set of simple time-dependent agents, the accuracy of a majority of the models actually decreased over time. Furthermore, in line with our previous paper on quality evaluation, we found that frequency and value models are the best, despite that the models have currently not received attention in literature. Finally, for each of the four types of opponent models we distinguished, we provided directions on how to improve the models.

Concluding, in this work we introduced a survey providing structure to the field of opponent modeling and introduced two methods which augment each other in quantifying the quality of an opponent model of the opponent's preferences. There are two straightforward directions for future work; first, the two evaluation methods could be combined in a single benchmark which can be easily used to quantify the quality of novel opponent models; second, a similar evaluation method can be created for the other five types of opponent models. We believe that both directions for future work are an important step in advancing the field of opponent modeling.

Appendix A

Implementation

The experiments discussed in the previous chapters were run using GENIUS [20]. GENIUS is a flexible program which can be used to simulate various types of negotiations. The original version of GENIUS did not implement all functionality required to run our experiments. Therefore, we extended the GENIUS in four ways: the BOA framework (Section A.1), Distributed Genius (Section A.2), quality measures (Section A.3), and general improvements (Section A.4).

A.1 BOA Framework

The BOA framework is a major extension of GENIUS which allows to separately develop the bidding strategy, opponent model, opponent model strategy, and acceptance strategy of a agent. Figure A.1 provides an overview of the components of an agent created using the BOA framework.

To add a BOA agent to a negotiation the user can use the GUI depicted in Figure A.2. In this GUI, a user can simply select a component of each type to create a new agent. Additionally, parameters can be given to each component, which can be used to test variants of the same component.

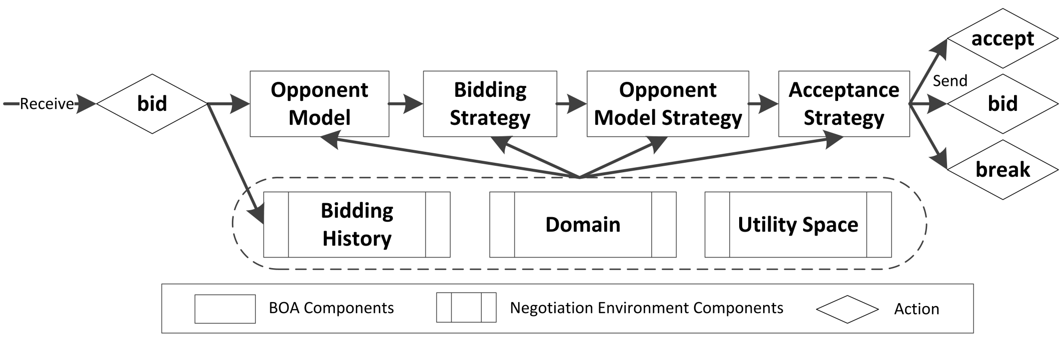

Figure A.1: Overview of the structure of the agent template used to create a BOA agent.

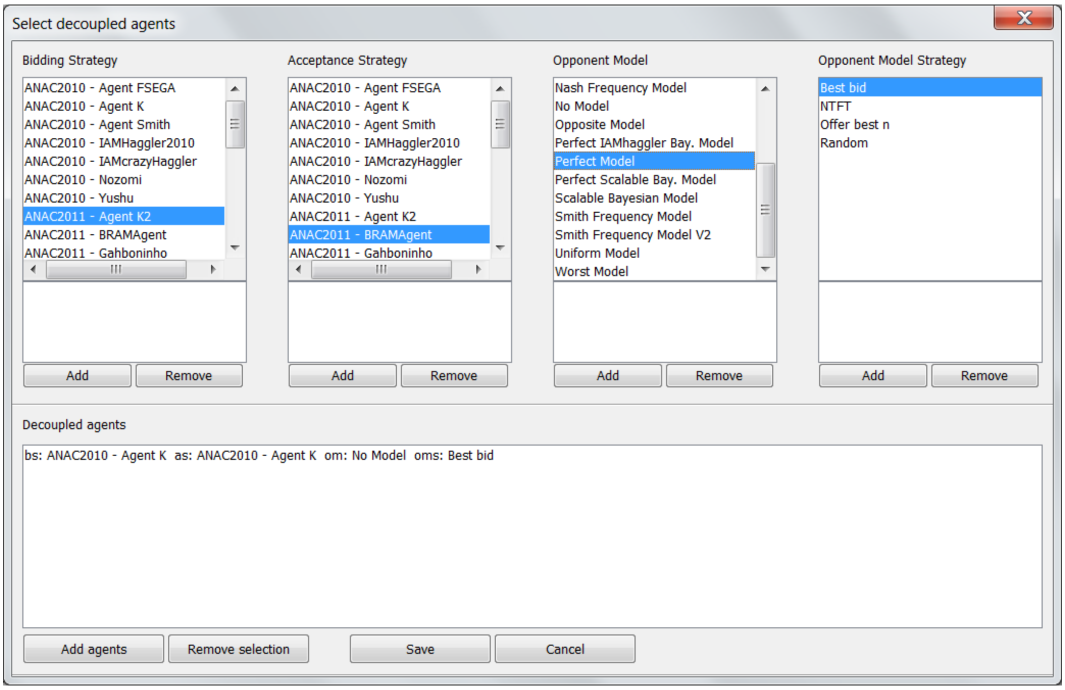

Figure A.2: The GUI of the BOA framework.

When a negotiation starts, the components of the BOA agent are loaded from file and packed as a normal negotiation strategy by copying them in the agent template illustrated in Figure A.1. This entails that from an outside perspective the BOA agent is similar to a normal agent and therefore we can benefit from existing functionality.

A new component can be easily created by implementing the interface of the type of component. Following, by adding the component to the repository, the component is added to the list of available components. We added a detailed explanation on how to use the BOA framework to the manual of GENIUS.

In contrast to the paper on the BOA framework, for the other papers the framework was extended in two ways:

1. First, a new component called opponent model strategy was added, which allows to specify how the bidding strategy uses the opponent model to select a bid. To illustrate, an opponent model can select the best bid from the set of similarly preferred bids.

   Furthermore, using this component, the default updating rule that all bids are used to update the opponent's model can be replaced. For example, an opponent model strategy can specify that the opponent model only updates half of the time on a large domain and thereby save computational resources.

2. Second, in line with the requirements for the ANAC2012 competition, the possibility to send a break action was added. This action immediately breaks off the negotiation, which may be beneficial in domains with discounts and a high reservation value.

Finally, we created a set of support classes to more easily create new bidding strategies. For example, a bidding strategy may benefit from using the *SortedOutcomeSpace* class to efficiently search the outcome space.

A.2 Distributed Genius

The problem with the default implementation of GENIUS, is that large tournaments can take days to complete on a single computer. Up till now, the solution was to manually split up the tournament, and let each computer run a part of the tournament. As we had to run a large number of tournaments, we created Distributed Genius.

Distributed Genius is an extension of GENIUS which can be used to divide a tournament among multiple computational threads, which may be different computers. Figure A.3 depicts an overview of how the system works. Initially, a user specifies the tournament to be run. The tournament is automatically split into smaller tournaments called jobs, which are stored in a central database together with the specification of the tournament. Subsequently, clients can join the tournament and are automatically allocated a job. When a job is finished, the results are stored in the database. Finally, when all jobs have been processed the results of the tournament are sent to all clients.

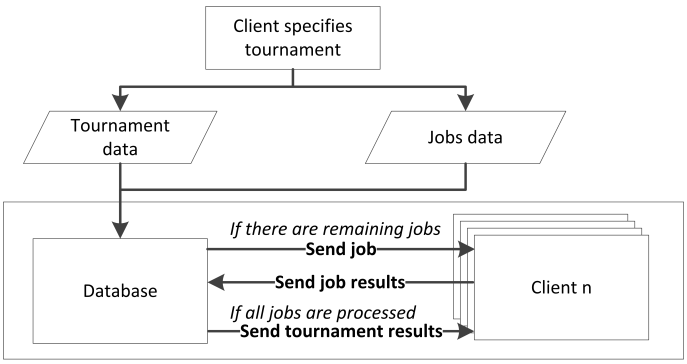

Figure A.3: Creating an running a tournament using distributed GENIUS.

During a tournament, it can happen that a computer crashes and therefore fails to finish its job. In this case, after all other jobs have been processed, the other clients detect that a job has been left unfulfilled an automatically start on this job.

In our configuration we used a small set of computers and a simple database server. At times, more than 20 computational threads were working on the same tournament, effectively realizing a speed-up of 20 times over using a single computer.

A.3 Quality Measures

Originally, GENIUS included only a small set of quality measures insufficient for our purposes. Therefore, we implemented five additional sets of quality measures. Each set is implemented in a way that it can be easily extended with novel quality measures.

1. *Opponent model accuracy measures* can be used to measure the accuracy of an opponent model during a negotiation. More than ten measures are included, which are discussed in detail in Chapter 5.
2. *Outcome measures* quantify the quality of the outcome of a single match and includes four measures: Kalai distance, Nash distance, Pareto distance, and social welfare.
3. *Scenario Measures* are a separate set of measures used to measure the properties of a scenario, for example the opposition of the preference profiles or their bid distribution. This set includes nine measures.
4. *Tournament measures* is a collection of more than twenty measures which summarize the results of a tournament. The set of measures include the average time of agreement, average rounds, and percentage of agreements.
5. *Trajectory measures* capture properties of the negotiation trace of an agent using measures from [10], such as the percentage of concessions. Besides the measures from [10], we included two measures which quantify the percentage of bids explored.

A.4 General Improvements to Genius

Besides the extensions of GENIUS discussed in the previous section, there are three major improvements we made to existing functionality:

1. *Memory leaks*. During tournaments we noted that the memory became full over time. This resulted in less offers being made, which in some cases strongly influenced the outcome of the negotiation. Using a memory analyzer we found several bugs which we resolved. The current version of GENIUS can run for days without problems.
2. *GUI*. In the original GUI, when a list of scenarios or agents exceeded a maximum length then they could no longer be selected. To resolve this problem, we generalized both GUI's to a single scrollable GUI which can be given an arbitrary list of items. Our improved GUI has been included in the last official builds of GENIUS.
3. *Pareto frontier*. During the implementation of the accuracy measures, we required a faster algorithm to calculate the Pareto frontier. Tim Baarslag had already written a faster algorithm, however, this version was not included in the main branch as it was not validated. Therefore, we created a test method which compares the results of an efficient algorithm with a slow bruteforce algorithm.

   Surprisingly, we found that both the new and the old algorithm returned incorrect results in a specific case in which two Pareto optimal bids were identical with regard to their utility for both parties. In large domains, a significant percentage of the Pareto bids have the same utility. In this case, Pareto bids were incorrectly discarded. This is a serious problem, as it can lead to biased results when the bids are used for example by the accuracy measures. Therefore, we adapted the faster algorithm by Tim Baarslag, which is incorporated in the latest builds of GENIUS.

Appendix B

Contribution of Authors

In line with the requirements, this section discusses my contribution to each paper. The order of the papers follows the structure of the main text. The survey of opponent models is not discussed, as the version in this thesis was fully written by me.

B.1 Decoupling Negotiating Agents to Explore the Space of Negotiation Strategies

Tim Baarslag proposed the idea to split the negotiation strategy into components to analyze the quality of individual components. I refined this idea by analyzing which components can be found in existing strategies. Based on an analysis of a large set of existing agents, I identified four components: the bidding strategy, acceptance strategy, opponent model, and opponent model strategy. The idea to add the opponent model strategy was one of my insights which I obtained from analyzing existing agents. For each component I defined an interface, and implemented a framework to combine different components. Based on this framework, at the end of my thesis, I implemented more than 80 such components in Java partly derived from more than 20 agents. As part of the first experiment, which was thought up by me and Alex Dirkwager, for all decomposed ANAC agents I validated whether the combination of components was equal to the original agent.

Furthermore, as part of the second experiment thought up mainly by Tim Baarslag, I tried to improve the state of the art by combining the best components. After more than a week without success, as a result of a discussion with Tim Baarslag, I executed and analyzed an alternative experiment in which I tried to improve every ANAC 2011 agent. I reported the results of this experiment in the paper.

Writing the paper was joint work by me, Alex Dirkzwager and Tim Baarslag. Tim Baarslag was the main author. Alex Dirkzwager and I wrote large parts of every sections, and Tim Baarslag wrote the outline of the paper, provided detailed feedback, and in general improved the quality of the writing. In total, I wrote nearly half of the work.

B.2 A Competitive Strategy for Real-time Bilateral Negotiations

As part of our master track, me and Alex Dirkzwager designed a negotiation agent as part of an artificial intelligence course. The agent competed in ANAC 2010 against state of the art negotiation strategies created by research teams all over the world. In this competition we entered the finals and ultimalely finished sixth. As part of our placement in the finals, we presented our work at AAMAS 2011. In addition, we wrote a paper about our negotiation agent which is published in the post-proceedings of the ACAN 2011 workshop.

Furthermore, during our master thesis we designed a new negotiation strategy which participated in the ANAC 2012. This negotiation strategy was created based on a detailed analysis of the best existing agents. The design and implementation of the agent was a joint effort. While Alex Dirkzwager spent relatively more time on the design, I focused on the efficient implementation of the agent. Furthermore, I wrote the paper, whereas Alex Dirkzwager provided feedback. Finally, I presented the agent at the ACAN 2012 workshop on the AAMAS 2012.

B.3 Measuring the Performance of Online Opponent Models in Automated Bilateral Negotiation

This paper concerns the evaluation of the performance of a model. To analyze and compare the quality of a set of opponent models, I came up with an experimental setup to fairly quantify the performance of an opponent model. For the experiment I implemented a large set of quality measures based on a survey of literature.

Furthermore, I ran the experiment and stored the results in an easily accessible format which I devised. Using this dataset, I tested a set of hypotheses using statistical methods. I documented my results, which were later on added to the paper.

Finally, writing the paper was joint work by me and Tim Baarslag. Tim Baarslag was the main author. He came up with the outline of the paper and improved the quality of the work in general. Similar to the BOA paper, I wrote large parts of the paper.

B.4 Measuring and Modeling what Matters in Automated Negotiations

This paper concerns the evaluation of the accuracy of a model. To evaluate the accuracy of a set of opponent models, I came up with an experimental setup to fairly quantify the accuracy of a model. This experimental setup contains the implementation of set of accuracy measures which I derived from literature; however, I also implemented a set of novel measures. My best accuracy measure proved to be better than all currently existing accuracy measures. I refined my experimental setup in discussion with my co-authors. In addition, I analyzed the results of the experiment and visualized the results using multiple figures.

In contrast with what is currently believed in the field of opponent modeling, I found that many opponent models degrade in accuracy over time. Furthermore, I found the most simple most to perform the best, despite that the more complex models – with poor performance – are popular in literature.

Tim Baarslag devised the second experiment in which the relation between performance and accuracy was analyzed. Similar to the first experiment, I created the initial experimental setup. Furthermore, I executed the experiment, and both analyzed an visualized the results.

Based on the results of both experiments, I concluded which type of model is best and how to improve the quality of each type of opponent model. For each type of model, I formulated detailed directions on how to improve the models and thereby improve the state of the art.

Similar to the other papers, the writing process was joint work by me an Tim Baarslag. Tim Baarslag is the main author of the work, and came up with the initial structure of the paper. In addition, he gave feedback on my writing. In its current form, the majority of the writing is done by me. After my thesis, we plan to finish the paper.

Bibliography

[1] Tim Baarslag, Katsuhide Fujita, Enrico H. Gerding, Koen Hindriks, Takayuki Ito, Nicholas R. Jennings, Catholijn Jonker, Sarit Kraus, Raz Lin, Valentin Robu, and Colin R. Williams. Evaluating practical negotiating agents: Results and analysis of the 2011 international competition. *Artificial Intelligence Journal*, Accepted.

[2] Tim Baarslag, Koen Hindriks, Catholijn M. Jonker, Sarit Kraus, and Raz Lin. The first automated negotiating agents competition (ANAC 2010). In *New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence*. Springer-Verlag, 2012.

[3] C. Beam and A. Segev. Automated negotiations: A survey of the state of the art. *Wirtschaftsinformatik*, 39(3):263–268, 1997.

[4] K. Binmore and N. Vulkan. Applying game theory to automated negotiation. *Netnomics*, 1(1):1–9, 1999.

[5] T. Bosse and C.M. Jonker. Human vs. computer behavior in multi-issue negotiation. In *Rational, Robust, and Secure Negotiation Mechanisms in Multi-Agent Systems*, pages 11–24. IEEE, 2005.

[6] R. Carbonneau, G.E. Kersten, and R.klaue2001automated Vahidov. Predicting opponent's moves in electronic negotiations using neural networks. *Expert Systems with Applications*, 34(2):1266–1273, 2008.

[7] R.M. Coehoorn and N.R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation trade-offs. In *Proceedings of the 6th international conference on Electronic commerce*, pages 59–68. ACM, 2004.

[8] EH Gerding, D.D.B. van Bragt, JA La Poutre, and Centrum voor Wiskunde en Informatica. *Scientific approaches and techniques for negotiation: a game theoretic and artificial intelligence perspective*. Citeseer, 2000.

[9] R.H. Guttman, A.G. Moukas, and P. Maes. Agent-mediated electronic commerce: a survey. *The Knowledge Engineering Review*, 13(02):147–159, 1998.

[10] K. Hindriks, C.M. Jonker, and D. Tykhonov. Let's dans! an analytic framework of negotiation dynamics and strategies. *Web Intelligence and Agent Systems*, 9(4):319–335, 2011.

[11] K. Hindriks and D. Tykhonov. Opponent modelling in automated multi-issue negotiation using bayesian learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems*, 2008.

[12] H. Jazayeriy, M. Azmi-Murad, M.N. Sulaiman, and N.I. Udzir. A review on soft computing techniques in automated negotiation. *Scientific Research and Essays*, 6(24):5100–5106, 2011.

[13] H. Jazayeriy, M. Azmi-Murad, N. Sulaiman, and N. Izura Udizir. The learning of an opponent's approximate preferences in bilateral automated negotiation. *Journal of theoretical and applied electronic commerce research*, 6(3):65–84, 2011.

[14] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, M.J. Wooldridge, and C. Sierra. Automated negotiation: prospects, methods and challenges. *Group Decision and Negotiation*, 10(2):199–215, 2001.

[15] R. Kowalczyk, M. Ulieru, and R. Unland. Integrating mobile and intelligent agents in advanced e-commerce: A survey. In *Agent Technologies, Infrastructures, Tools, and Applications for E-Services*, pages 295–313. Springer, 2002.

[16] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94(1-2):79–97, 1997.

[17] S. Kraus. Automated negotiation and decision making in multiagent environments. *Multi-agent systems and applications*, 1:150, 2001.

[18] C. Li, J. Giampapa, and K. Sycara. A review of research literature on bilateral negotiations. Technical report, Robotics Institute, Pittsburgh, PA, November 2003.

[19] R. Lin, S. Kraus, J. Wilkenfeld, and J. Barry. Negotiating with bounded rational agents in environments with incomplete information using an automated agent. *Artificial Intelligence*, 172(6-7):823–851, 2008.

[20] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M. Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 2012.

[21] Alessio R. Lomuscio, Michael Wooldridge, and Nicholas R. Jennings. A classification scheme for negotiation in electronic commerce. *Group Decision and Negotiation*, 12:31–56, 2003.

[22] Christoph Niemann and Florian Lang. Assess your opponent: A bayesian process for preference observation in multi-attribute negotiations. In Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo, editors, *Advances in*

Agent-Based Complex Automated Negotiations, volume 233 of *Studies in Computational Intelligence*, pages 119–137. Springer Berlin / Heidelberg, 2009.

[23] Martin J. Osborne and Ariel Rubinstein. *Bargaining and markets*. Academic Press, 1990.

[24] Y. Oshrat, R. Lin, and S. Kraus. Facing the challenge of human-agent negotiations via effective general opponent modeling. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 377–384. International Foundation for Autonomous Agents and Multiagent Systems, 2009.

[25] Yong quan Liang and Yong Yuan. Co-evolutionary stability in the alternating-offer negotiation. In *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1176–1180, 2008.

[26] H. Raiffa. *The art and science of negotiation*. Belknap Press, 1982.

[27] J.Z. Rubin, B.R. Brown, and M. Deutsch. *The social psychology of bargaining and negotiation*. Academic press, 1975.

[28] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica: Journal of the Econometric Society*, 50:97–109, 1982.

[29] G.C. Silaghi, L.D. Serban, and C.M. Litan. A framework for building intelligent sla negotiation strategies under time constraints. In *Proceedings of Economics of Grids, Clouds, Systems, and Services: 7th International Workshop*, volume 6296, page 48. Springer-Verlag New York Inc, 2010.

[30] D. Zeng and K. Sycara. Bayesian learning in negotiation. *International Journal of Human-Computers Studies*, 48(1):125–141, 1998.