Designing and Building a Negotiating Automated Agent

Sarit Kraus*
Department of Mathematics and Computer Science
Bar-Ilan University
Ramat Gan 52900, Israel
e-mail: sarit@bimacs.cs.biu.ac.il (972)-3-531-8407

Daniel Lehmann
Institute of Computer Science,
Hebrew University, Jerusalem 91904, Israel
lehmann@cs.huji.ac.il (972)-2-58-5258

Note: This article appears in *Computational Intelligence*, 11(1):132-171, 1995.

*Sarit Kraus is affiliated also with the Institute of Advanced Computer Studies at the University of Maryland.

Abstract

Negotiations are very important in a multi-agent environment, particularly, in an environment where there are conflicts between the agents, and cooperation would be beneficial.

We have developed a general structure for a *Negotiating Automated Agent* that consists of five modules: a Prime Minister, a Ministry of Defense, a Foreign Office, a Headquarters and Intelligence. These modules are implemented using a dynamic set of local-agents belonging to the different modules.

We used this structure to develop a Diplomacy player, *Diplomat.* Playing Diplomacy involves a certain amount of technical skills as in other board games, but the capacity to negotiate, explain, convince, promise, keep promises or break them, is an essential ingredient in good play. *Diplomat* was evaluated and consistently played better than human players.

Key Words: Automated Negotiations, Multi-Agent Environment, Game Playing, Local-agents, Diplomacy.

Subject Category: Cognitive Science, Knowledge Representation.

1 Introduction

Negotiation is an effective way of arriving at understanding and achieving cooperation between different entities: people, organizations and countries. We negotiate with the members of our family, our landlord, our boss at work and the salesperson at the shop. Unions negotiate with companies. Statesmen negotiate in order to improve the situation of their countries. Businessmen negotiate in order to reach a lucrative deal.

The need for negotiation also arises in a multiagent environment where cooperation between the agents may be beneficial; it becomes even more crucial in the presence of conflict among the agents.

We have developed a general structure for a *Negotiating Automated Agent* that consists of five modules: the Prime Minister, the ministry of Defense, the Foreign Office, the Headquarters and the Intelligence. These modules are implemented by a dynamic set of local-agents that work together, communicate and exchange messages to achieve the common general tasks of the *Negotiating Automated Agent*.

As a testbed, a specific domain was chosen, rich enough to include most aspects of negotiation. Given a (restricted version of) natural language which covers this domain, our agent was confronted with human agents and even showed an advantage over its human negotiation partners.

In the following section we will describe the community of agents we are considering. Then we will describe the game of Diplomacy (Section 3.1), which is the environment for which the *Negotiating Automated Agent*, *Diplomat*, was developed. We will explain the importance of negotiation for success in Diplomacy, and in Sections 3.2 and 3.3 we will describe the structure of *Diplomat*. We conclude this Section (3.4) with a detailed example of the internal structure of *Diplomat* during a specific Diplomacy game. In section 4 we will describe the experiments we conducted with *Diplomat*, followed by a discussion (Section 5) of the general structure of the *Negotiating Automated Agent*. In Section 6 we discuss related work in Distributed Artificial Intelligence and compare our structure of the *Negotiating Automated Agent* with other related structures. We will conclude in Section 7 and describe some possible research directions.

2 The Agents Community

We assume that there is a set of autonomous agents acting in a dynamic environment. Each agent has a set of general goals it needs to fulfill. The agents are autonomous; they have their own utility functions, and no global notion of utility plays a role in their design. The agents are *individually motivated*. The set of agents may change over time as a result of changes in the environment.

Cooperation between the agents may be beneficial but conflicts among the agents can arise. There is no mediator and central controllers do not exist, so the agents must coordinate their actions through communication and negotiation.

We assume that the agents have incomplete information concerning the other agents' goals and tasks, and an agent can provide the other agents with false information. Furthermore, since there is no central control, agreements are *not* enforced and agents may break their promises.

The agents must have a common language. To make the automated negotiations a bit easier, we have developed a Negotiation Language that can be used by both automated and human agents. The Negotiation Language includes four kinds of messages: declarations, questions, suggestions and answers (see [Kraus 1988]). The building blocks for messages are simple sentences that are specific to the subject of the negotiations. However, the structure of the messages are general. Currently, the basic components of the Negotiation Language are specific to Diplomacy, but can be changed to be used in any related environment. We have shown that the language is rich enough for diplomacy negotiation by translating all the messages that were exchanged in a real Diplomacy game (played by humans) and were written in natural English to the formal Negotiation Language.

To enable the Negotiating Automated Agent to use the language easily it sends and receives "Lisp" like messages. To make it easier for the human agents to negotiate using the Negotiation Language, they use "English" like messages. A special directed editor that was developed[1] helps the human agents use the formal negotiation language and transfers the English-style representation into the Lisp-style representation and vice versa.[2]

The main problem in developing a *Negotiating Automated Agent* that will act in such an environment is that the tasks that this agent must perform are varied and changing

[1]The special editor was developed by Arie Schlesinger [Schlesinger 1987].

[2]We haven't put much effort in translating the Lisp-style syntax into natural language, and that explain the very awkward construction of the English-style sentences in the examples.

over time. An agent must analyze the situation, determine its current tasks in order to achieve its general goals, and plan its activities. It has to choose with whom to negotiate and when. Then it must decide on its tactics for the negotiations and be able to evaluate offers and suggestions. It has to decide whether to sign the proposed agreements and then must determine which of them to follow and which to break. Finally, it must perform its activities.

To build an implementation of an automated agent becomes even more complex when the agent holds simultaneous negotiations. Negotiating with only one party at a time greatly simplifies the system, but the agent loses opportunities with other agents while dealing with only one of them (e.g., they may reach an agreement opposing him). By negotiating with only one agent at a time, it also cuts itself off from information which it might have received while negotiating with other agents. In addition, if not responding to offers from another agent, its silence may be interpreted as a negative response. Therefore, simultaneous negotiation with all parties is beneficial.

In order to overcome the problems that arise while negotiating in such a complex and dynamic environment, we have developed a structure which can adapt to these changes. We have used these ideas for building a successful automated Diplomacy player.

There are two main factors that contribute to Diplomat's success (as we will describe in Section 4). The first one is the detailed algorithms that are used for solving different problems arising during the negotiation such as: with whom to negotiate, evaluation of suggestions and the way to offer suggestions. In a complementing paper [Kraus *et al.* 1991] we described the algorithms that *Diplomat* uses to solve these problems.

The other factor of *Diplomat*'s development is its internal structure that allows it to act and negotiate simultaneous with several other players in a changing environment with incomplete information. We describe these aspects of *Diplomat* in this paper.

3 Diplomacy and *Diplomat*

3.1 Diplomacy

Diplomacy, is a board game marketed by Avalon Hill Company.[3] Diplomacy is an environment of intense negotiation. We implemented and tested our ideas by building an automated

[3]Similar description of Diplomacy (Section 3.1) and a brief description of *Diplomat* (Section 3.2) appear also in [Kraus *et al.* 1991]. Preliminary results appeared in [Kraus 1988; Kraus *et al.* 1989; Kraus and Lehmann 1991].

Diplomacy player called *Diplomat*.

Diplomacy is a board game played on the map of Europe during the years just prior to World War I. (See Figure 1). Each player represents one of seven European powers: England, Germany, Russia, Turkey, Austria-Hungary, Italy and France. The countries are depicted on the board by heavy, solid black lines (See Figure 1); the Great Powers are also subdivided into provinces. The seas are divided into "bodies of water". Certain provinces on the board are designated "supply centers." The supply centers are important places in Europe and are marked by black dots on the map (see Figure 1; for example, London and Berlin are supply centers). Each of the supply centers produces supplies needed to maintain an army or fleet. A player may have only as many armies and fleets on the board as the number of supply centers he controls.

The object of the game is for one power to gain control over the majority of the board, i.e., to control 18 supply centers. Since gaining control of the board can be time consuming, the game can also be played with a predetermined number of moves. The winner in such an abbreviated game is the player with the most units on the board. The stages of the game are divided into two seasons a year: a Spring season and a Fall season, beginning in the year 1901.

Coalitions and agreements among the players significantly affect the course of the game. Agreements include mutual promises for cooperation; an agreement may be detailed as to contain detailed military cooperation. The coalitions and agreements are determined during the diplomacy periods which take place before each move. During these diplomacy periods a player may say anything s/he wishes[4]. The communications usually consist of bargaining or joint military planning, but they may also include exchanges of information, denounciations, threats, rumors, etc. The communications between players are usually secret to the other players. The rules of the game do not predetermine which powers need to be allies or enemies; cooperation is reached only through negotiations. The rules of the game do not bind a player to anything she says. Deciding who to trust as situations arise is part of the game.

At the end of the negotiation period, all players secretly write the orders for all of their units simultaneously. A unit may be ordered to do only one thing in each season: to hold, move, or provide support. A fleet may also be ordered to convoy an army from one coast to another. Only one unit may be in any space at one time. Therefore, an army (resp. a fleet) may move to any adjacent province only if this move does not cause it to conflict with

[4]In the games we conducted, we restricted the negotiations to a special formal negotiation language that we have developed; the negotiations were conducted through electronic mail.

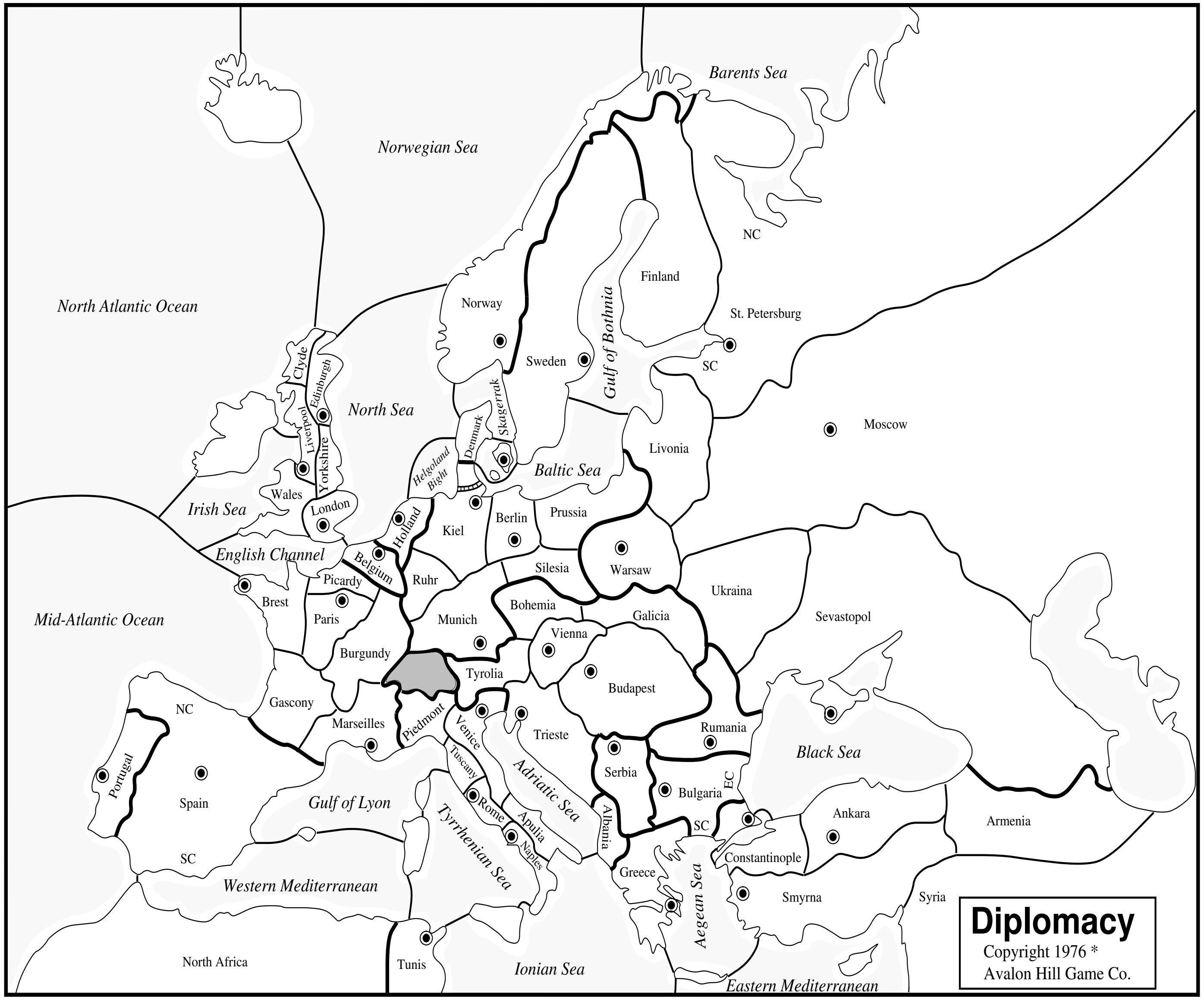

Figure 1: Diplomacy's map

another unit.

An army or a fleet may support the move of another army or fleet of that country or any other country in making a move. Support can also be given on a defensive basis to protect a playing piece from being dislodged from the space it occupies.

If one piece has the support of another in making the move and the opposing unit does not, then the supported unit moves into the empty space — or it may force an unsupported unit to retreat from a space. To "support" a move, the supporting unit must itself be able to move into the space under attack. Opposing units with equal support do not move. An advantage of only one support is sufficient to win.

The number of units that a player has on the board is determined by the number of "supply centers" which s/he controls. Adjustments in strength are made after the Fall moves have been completed. Each player writes his desires. If he increased the number of supply centers under his control, he writes whether he wants an army or a fleet and , if so, where. The unit must be placed in a supply center in his home country. If he has lost strength, he decides which army or fleet is to be removed. The following example demonstrates some of the rules of Diplomacy. Further details of the rules of the game can be found in [The Avalon Hill Game Co. 1976].

Example 3.1 It was Spring 1902. The game board situation appears in Figure 2. We will examine part of the orders that were given:

England:
1. F North Sea Convoys A Yorkshire to Belgium
2. A Yorkshire moves to Belgium
3. F London moves to English Channel

Italy:
4. A Venice moves to Trieste
5. A Bohemia moves to Vienna
6. A Tyrolia supports A Bohemia to Vienna
7. F Tunis moves to Ionian Sea

Austria:
8. A Serbia moves to Trieste
9. A Budapest moves to Vienna
10. A Trieste moves to Venice

Power	Supply Centers	Units
Russia (*Diplomat*)	Warsaw Sevastopol Vienna Rumania Sweden St. Petersburg Moscow	A St. Petersburg, A Rumania, A Vienna A Warsaw, A Moscow F Sevastopol, F Sweden
Austria	Serbia Trieste Budapest	A Budapest, A Serbia F Trieste
England	Norway London Liverpool Edinburgh	F London, F North Sea, F Norway, A Yorkshire
France	Portugal Spain Brest Marseilles Paris	A Paris, F Portugal, A Burgundy A Spain, A Marseilles
Germany	Kiel Berlin Munich Holland Denmark	A Holland, A Berlin, F Denmark A Munich, A Kiel
Italy	Naples Venice Tunis Rome	A Bohemia, A Tyrolia A Venice, F Tunis
Turkey	Bulgaria Greece Ankara Constantinople Smyrna	F Smyrna, F Ankara, A Greece A Bulgaria, A Constantinople,

Figure 2: A situation in Spring 1902: *Diplomat* plays Russia. A Supply centers are places that produce supplies sufficient to keep an army or a fleet going. Units are armies or fleets. We denote an army with A and and a fleet with F.

Russia:

11. A Rumania supports Vienna to Budapest
12. A Vienna moves to Budapest

Results:

England's army in Yorkshire entered Belgium (1,2) and England's fleet in London entered the English Channel. Italy's attempt to enter Trieste (4) failed due to a similar attempt by Austria to enter Trieste from Serbia (8) with equal forces. Italy's army in Bohemia entered Vienna (5) despite Austria's attempt to also enter Vienna from Budapest. The success was due to the support by the army in Tyrolia (6). Italy's fleet in Tunis entered the Ionian Sea (7). Austria's attempt to enter Venice (10) failed since Italy's army in Venice stayed. Russia's army in Vienna entered into Budapest with the support of the army in Rumania (11,12).

Although each player in Diplomacy tries to win the game, players can benefit from cooperation; there are cooperative strategies that can improve both players' situations, i.e., if there is a strong player, who no other player can defeat by himself, his opponents can work together against him, gain control over his supply centers and share them amongst themselves.

The need for negotiation increases when certain moves require close cooperation between various allied powers. The units of one power may support the moves of another power only if they are explicitly ordered to do so. For example, if Russia wants to help Italy to enter Trieste (as in example 3.1), it must specify the unit that will perform the incursion.

The game is quite complex; there are 75 spaces on the board that can be occupied by 34 units. At the end of each season each unit is ordered simultaneously to perform one of four possible actions (move, support, convoy, hold). All of the actions beside "hold" are related to the spaces adjacent to the space that the unit occupies (usually about 5 spaces). We conclude that there are approximately 34^{16} possible moves in each step of the game.

Since the game is so complex, it is difficult to evaluate the preferences of the other players given different possible board game situations. It is even harder to estimate the preferences of the players given different possible strategies. Therefore, a Diplomacy player negotiates in order to obtain information about the goals of the other players. He also tries to receive information from one player on her evaluation of the goals and strategies of a third party.

The need for negotiation increases in Diplomacy since other players negotiate. The negotiations between other players may lead to coalitions against a player, and if s/he does not negotiate, s/he can find himself without any help against those players. Experience shows that a power cannot last long without taking part in extensive negotiations.

In Diplomacy each player represents a different country, and therefore the initial prospects, problems and positions of each player are different. Nevertheless, all players have almost equal strategic chance to win the game. To be a successful Diplomacy player one needs some technical skills in moving military units on the board according to the reasonable but complex rules of the game, but more importantly one needs the ability to communicate and negotiate with the other players, to make agreements and possibly, to ignore prior agreements.

There are several common attributes that are important for a negotiator regardless of the environment in which s/he negotiates (see [Karrass 1970]). We assume that a structure that is applicable and useful to an automated Diplomacy player will be applicable in similar environments in which negotiation is an important issue.

3.2 *Diplomat*'s General Structure

The general structure of *Diplomat* can be described as a government which includes a Prime Minister, a Ministry of Defense, a Foreign Office, a Military Headquarters and Intelligence[5].

The Prime Minister directs *Diplomat*'s activities. Its personality influences the behavior of the government's members. The Prime Minister has a secretary that maintains the government's knowledge base. The Ministry of Defense is responsible for planning and situation analysis; it directs the behavior of the Foreign Office, the Military Headquarters and the Intelligence. The Foreign Office is responsible for communicating with the other powers. The Military Headquarters decides on *Diplomat*'s moves at the end of each season, and Intelligence acquires information about the other players. Another module is the Strategies Finder (SF) which provides strategies[6].

The following is a short description of the modules (see Figure 3):

- The Prime Minister - directs *Diplomat* activities. It is the only module that has 'personality' traits such as aggressiveness, willingness to take chances and loyalty. The 'personality' traits are given to it at installation or during the negotiation. They allow *Diplomat* to change 'personality' from one game to another, and it is therefore more difficult for other players to estimate its intentions.

  The government's secretary, which is directed by the Prime Minister, keeps and maintains *Diplomat*'s Knowledge and Belief Base according to the information which flows from all the other modules. The Knowledge and Belief Base (KBB) includes the following parts:

  - Diplomacy's rules (including general information about the board).
  - The situation of the game (see an example in Figure 2 above).
  - The messages that were exchanged between *Diplomat* and other players.
  - Agreements table (see an example in Figure 4 below).

[5] *Diplomat* is implemented in Ylisp [Levy and Dimitrovski 1984] (a dialect of Franz-Lisp) on a Vax 11/785 running Unix, Berkeley 4.3. It was developed and tested in 1984-1989.

[6] We define a strategy, in the Diplomacy game, to include a list of orders and the expected loss or profit from it. Each order includes the active unit, the type of activity (Move, Support, Convoy and Hold), and the locations on the board that are related to this activity. The purpose indicates whether the action is part of a plan to attack a specific area or part of a plan to defend a specific area. For example, "The Rumanian army in Bulgaria moves to Greece in order to attack Greece". The strategy also includes the expected average profit from carrying out the strategy for each power who participates in the strategy (i.e., at least one of its units is active in the strategy), and the common expected profit for all of the powers.

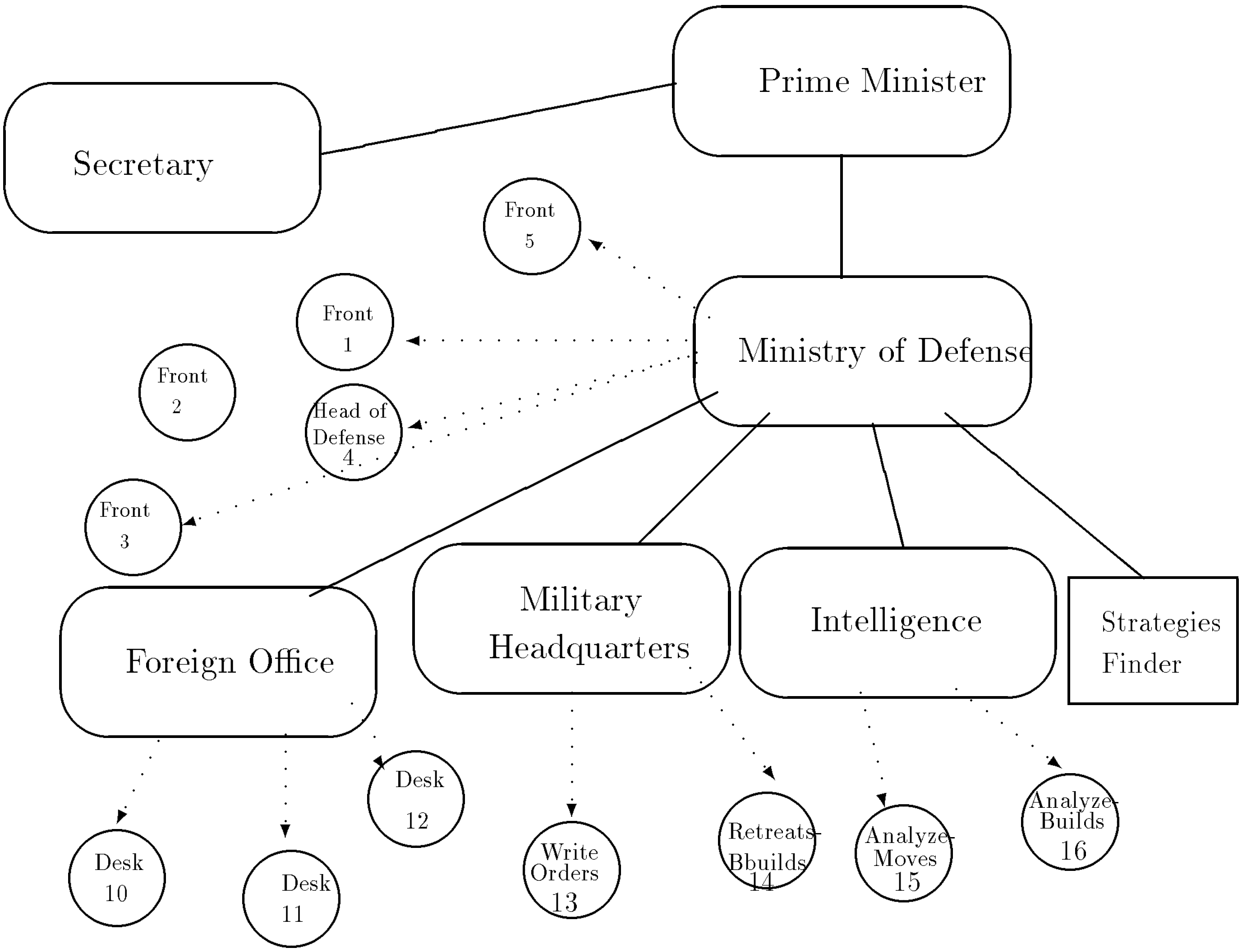

Figure 3: *Diplomat*'s general description

- Information about the other powers and their relations (see an example in Figure 5 below).

  Details about the KBB and the way it is maintained can be found in [Kraus 1988; Kraus and Lehmann 1991].

- Ministry of Defense - This module is responsible for the planning and analysis. It is influenced by the Prime Minister's personality traits. It directs the behavior of the Foreign Office, Headquarters and Intelligence and it uses the strategies finder for finding strategies. While analyzing a new game board situation, the Ministry of Defense considers different possible fronts[7] and coalitions and searches for possible strategies.

  An important mission of this module is to find partners for negotiation. A 'Desk' at the Foreign Office is established for each negotiation partner, and this Desk is responsible for the negotiations with this partner. The Ministry of Defense provides the 'Desk' with a strategy that can be a basis for negotiation with this partner. The Ministry of Defense also helps the Foreign Office to evaluate different suggestions received from the other powers.

- Foreign Office - This module directs the communications of *Diplomat* with the other powers. It includes different departments ('Desks') that are responsible for the relations with the different powers. These Desks conduct negotiation according to strategies received from the the Ministry of Defense. During the negotiation, it parses and analyzes messages, evaluates suggestions received from the another powers and makes suggestions to the other powers.

  It also decides whether to sign an agreement. After signing an agreement, it announces the details to the Prime Minister, and its secretary updates the agreements table.

- Military Headquarters - This module decides on *Diplomat*'s moves at the end of each season. It uses *Diplomat*'s KBB to find out who are its allies and its enemies (according to the agreements). It also gathers information with regards to the promises that *Diplomat* extended to its allies and decided to keep, and the activities *Diplomat*'s allies promised to carry out and *Diplomat* estimated what they will really perform. Military Headquarters asked the Strategies Finder to provide it with strategies that fit the set

[7] A **Front** includes possible enemies and possible allies. For example: Russia and Turkey against Austria and Germany or *Diplomat* against Italy.

Agreement	Allies	Enemies	Intention of *Diplomat*	Intention of the Ally	Details
1	Russia England		4	4	England won't enter into Barents Sea and St. Pet.
2	Russia Germany		4	4	Russia and Germany won't enter into Baltic Sea, Silesia, Bohemia and Prussia
3	Russia Turkey	Austria	5	5	Russia and Turkey won't enter into Armenia, Black Sea

Figure 4: *Diplomat*'s agreements table at the beginning of Spring 1902. The intention of *Diplomat* and its allies are on a scale between 0 and 10.

of allies and enemies and include the above promised activities. Military Headquarters chooses one of these strategies according to *Diplomat*'s personality and determines *Diplomat*'s moves.

- Intelligence - This module tries to estimate the relations between the powers and their characters. The estimation is primarily based on analysis of the powers' orders at the end of every season. *Diplomat* may also receive messages concerning another power's character and behavior or concerning the relations between other powers. Intelligence decides whether or not to believe the messages that were received. It reports its conclusions to the Prime Minister and its secretary updates *Diplomat*'s KBB.

- Strategies Finder (SF)[8] - Takes a set of allies, a set of enemies and the configuration of the board as an input. Based on this input it finds some plausible strategies and provides different values that serve for their evaluation. The Ministry of Defense, Headquarters and Intelligence use this module. They can influence the generation of strategies by providing the SF with portions of strategies that where already agreed upon (due to existing agreements), areas of special interest and various degrees of willingness to take chances. The SF was developed, based on observation and interrogation of human Diplomacy experts in the spirit presented by PARADISE [Wilkins 1983].

Before describing in detail the internal structure of *Diplomat*, let us describe a typical behavior of *Diplomat* during a season in the Diplomacy game.

[8]This module was developed by E. Ephrati [Ephrati 1987].

Power	Friends	Enemies
Russia	Turkey England Germany	Austria Italy
Austria		Russia
England	Russia	France
France		England
Germany	Russia	
Italy		Russia
Turkey	Russia	

Figure 5: *Diplomat*'s beliefs about the relations between the powers in the beginning of Spring of 1902

At the beginning of a season *Diplomat* analyzes the situation and looks for possible partners for negotiation (this is the Ministry of Defense's responsibility). Then it negotiates with these partners. The negotiation may be in order to reach a beneficial agreement, to prevent the formation of an opposing coalition or to gather information. More negotiations may be initiated during the season (this is the Foreign Office's responsibility). If an agreement is signed, its details are recorded in the agreements table (this is the Prime Minister responsibility). At the end of the season *Diplomat*'s orders are given taking into account the agreements signed until then (this is the Military Headquarters' responsibility). After the orders of the players are made known, *Diplomat* analyzes them in order to estimate the characters of the other players and the relationships between them (this is the Intelligence responsibility).

One may ask whether *Diplomat* plays the same way under the same circumstances, and whether the same local-agents will be created during the same situation[9]. The answer is no. The nondeterminism of *Diplomat* is a result of the following features. First, *Diplomat*'s Prime Minister has special "personality" traits that affect the general behavior of *Diplomat* and may be varied easily from game to game. In addition, *Diplomat* "flips coins" in the following cases:

- A desk in the Foreign office "flips" a coin with a probability that is influenced by the "personality" traits, to decide whether to pretend to keep an agreement or to tell the other partner that it will break the agreement,

- A desk in the Foreign office "flips" a coin to decide whether to give more details about

[9] A good negotiator should act in ways that are unexpected by its partners-adversaries.

a suggestion.

- In the beginning of the game, a coin is flipped by a department of the Ministry of Defense, to choose which opening to start with (see discussion in the end of Section 3.3.2).

- *Diplomat*'s Strategies Finder "flips" a coin when it searches for possible strategies. For example, to decide which units will participate in the attack or defense of a given location and to guess which of the enemy's units will participate in the battle of that location.

We feel that any development of a *Negotiating Automated Agent* using the interacting local-agent framework in the future will involve some non determinism.

3.3 *Diplomat*'s local-agents

Diplomat acts in a dynamic environment and all its modules need to perform different tasks. Some of the tasks are changing over time. For example, the Foreign Office needs to negotiate with several players. The Ministry of Defense should deal with different fronts, and Headquarters needs to gives *Diplomat*'s moves and decide on retreats and builds. While the structures of the tasks of each module are similar, the details vary from one specific task to another.

We have developed a framework for a dynamic set of local-agents[10] that belong to the different modules of *Diplomat*. For example, there is a set of local-agents that belongs to the Foreign Ministry. Each of these local-agents serves as a delegation for negotiation with one other player. These local-agents use similar procedures such as making offers or evaluating a suggestion. But, the set of delegations may change over time and therefore the set of local-agents that performs this mission may change over time. A different set of local-agents plan and analyze specific problems with which the Ministry of Defense should deal. For example, a local-agent may be responsible for a specific front. Here again, since the problems are changing over time, new local-agents may be created while others may vanish when they are no longer needed. Similarly, for Headquarters and Intelligence.

[10]We denote these agents as "local-agents" to distinguish them from the autonomous agents that belong to the agents community (Section 2). We note that we have two levels of distributed agents. At the *upper* level, autonomous agents act in a multi-agent environment (some of them may be humans). They are individually motivated and negotiate with each other to reach a mutually beneficial agreement. At the *lower* level, we developed an internal structure for an autonomous agent (our *Negotiating Automated Agent*), which consists of a dynamic set of local-agents that communicate and work together (but they don't negotiate) to achieve the common general task of the *Negotiating Automated Agent*.

All the local-agents of the same type, have similar tasks to perform and use similar procedures to do so. When a local-agent is created, it is given certain parameters (the kinds of parameters depend on the type of task) which define the details of its task and differentiate it from other local-agents of the same type.

Each local-agent acts independently, but a local-agent may be assisted by other local-agents either asking for information or delegating some of its work to the other local-agents, in order to complete one of its tasks. The local-agents use an internal-mail system and a notice board for communication.

The internal-mail system is used for communication between two local-agents that are acquainted with each other for a specific purpose. For example a local-agent that serves as a delegation for negotiation with a given player may send an internal message to another local-agent that belongs to the Ministry of Defense and is responsible for the planning of the issues that appear in the negotiations between *Diplomat* and the given player. The identity of the Ministry of Defense's local-agent should be known to the delegation local-agent.

If a local-agent that belongs to the ministry of Defense has found a solution for a given problem, and wants all the other local-agents to know how well *Diplomat* can solve the problem, it can post the solution on the notice board. The local-agents may also use the notice-board to broadcast information that is important to all of the local-agents, such as information related to resource allocation among the local-agents. The ability to use a notice board to broadcast messages is important since the set of local-agents changes over time. Therefore, sending a message to all the local-agents in existence when the message is created does not guarantee that it reaches all the agents that need to get it.

We have developed a *controller* that belongs to the Prime Minister. The controller manages the behavior of all the local-agents. It also sends and receives messages from the other players.[11]

When the controller receives a message from another agent it allocates the message to the local-agent that is most suitable to handle it. If such a local-agent does not exist, a new one is created. The controller chooses the local-agent that will be active and may pass internal-messages[12] between the local-agents. Local-agents may be created by the controller.

[11]The framework that we described assumes that there is only one processor that is available for the automated agent. If a distributed computer architecture is available, each local agent can work on a different processor with a local memory. This will avoid the need for choosing the active local-agent, and will of course make the *Negotiating Automated Agent* more efficient. Its response time to other agents' messages will be shorter. The problem will be to manage the scheduling of the processors. This is left for future work.

[12]We use the notion of "internal-message" to distinguish between the messages that are exchanged between

3.3.1 The Structure of Local-Agents

We have developed the structure of the local-agent according to the needs that arose in the implementation of a *Diplomacy* player. Since *Diplomacy* is a typical negotiation situation in a complex multi-agent environment, we assume that it will be useful in similar environments. The structure of each of the local-agent is as follows:

1. A unique **identifying number** which is used for addressing messages.

2. The **priority** of the local-agent. It is used by the controller to determine which local-agent will be activated next. It is computed when the local-agent is created and it is a function of the type of the local-agent and its parameters. For example, the priority of a local-agent that actually negotiates with another player is higher than other local-agents.

3. The **state** of the local-agent which is either *Run* or *Sleep*. A local-agent sleeps when it is waiting for an event (a message, for example). The controller chooses the next active local-agent from the local-agents that are in the *Run* state.

4. The **type** of the local-agent. There are various types of local-agents which perform different missions; for example, a local-agent that is responsible for the negotiations with a specific agent or a local-agent that makes plans.

5. The **parameters** of the local-agent. Each type of local-agent requires specific parameters, i.e., a local-agent that is responsible for the negotiation with another agent needs the name of that agent, and the local-agent that looks for a plan to solve a problem needs to know the objects that can be used to solve the problem.

6. The identifying numbers of the other local-agents with which the local-agent communicates. A local-agent usually sends its internal messages to the local-agents on this list.

7. **Mailbox** - the controller broadcasts the internal-messages among the local-agents and puts the relevant internal-messages in the mailbox.

8. **Private storage** where the local-agent keeps information it computes or collects. For example, a local-agent that is responsible for the negotiations with another agent keeps

---

the local-agents and the messages that are passed among the autonomous agents.

in its private storage the messages that were exchanged with this agent. A local-agent that is part of the Ministry of Defense keeps strategies that it computed in its private storage.

Diplomat has different types of local-agents. Usually, there exist several local-agents of the same type simultaneously with different parameters. A detailed example of *Diplomat*'s local-agents during a specific diplomacy period can be found in Section 3.4.

The motivation behind defining the following types was to establish a type of local-agent for any task a Diplomacy player must perform. The types are as follows:

Head-of-Defense: the head of the Ministry of Defense. Creates the different Fronts in the Ministry of Defense at the start of each season and allocates Diplomat's units among them.

Front: part of the Ministry of Defense. Responsible for a specific Front. (See an example in Figure 6[13]).

Desk: part of the Foreign Office. Negotiates with one of the powers (see an example in Figure 7.)

Write-Orders: part of the Military Headquarters. Plans and announces *Diplomat*'s moves at the end of each season.

Analyze-Moves: part of Intelligence. Analyzes the moves that made by the other powers at the end of each season and reports conclusions to the Prime Minister.

Analyze-Messages: part of Intelligence. Analyzes declarative messages received from other players. It reports its findings to the Prime Minister.

Retreats-Builds: part of the Military Headquarters. Writes *Diplomat*'s retreats, builds or disbands when needed.

Analyze-Builds-Retreats: part of Intelligence. Analyzes the builds, retreats and disbands of the other powers.

Open: part of the Ministry of Defense. It acts at the beginning of the game (i.e., Spring 1901).

<table>
<tr><th>Id</th><th>Priority</th><th>Status</th><th>Type</th><th>Parameters</th></tr>
<tr><td>6</td><td>10</td><td>Run</td><td>Front</td><td>Allies: Italy and Russia
Enemies: Austria and Germany
Type: Start
Friendly flag: nil
Units not to use: F Tunis</td></tr>
<tr><td colspan="5">Local Storage
....
A Venice (I) moves to Trieste in order to attack Trieste
A Vienna (R) supports A Venice to Trieste in order to attack Trieste
A Bohemia (I) moves to Munich in order to attack Munich
A Tyrolia(I) supports A Bohemia to Munich in order to attack Munich
A Rumania (R) supports A Warsaw to Galicia in order to attack Galicia
A Warsaw (R) moves to Galicia in order to attack Galicia
F Sevastopol (R) supports A Rumania
Expected outcomes: Aver:10617 Min: 5002 Max: 20862
Russia: 3358 Italy: 18117
...
A (R) Vienna supports Tyrolia to Trieste in order to attack Trieste
A (R) Rumania moves to Serbia in order to attack Trieste
A Tyrolia (I) moves to Trieste in order to attack Trieste
A Bohemia (I) supports Warsaw to Galicia in order to attack Galicia
A Venice (I)supports Tyrolia to Trieste in order to attack Trieste
A Warsaw (R) moves to Galicia in order to attack Galicia
F Sevastopol (R) moves to Rumania in order to defend Rumania
Expected outcomes: Aver:4937 Min: -3655 Max:10121
Russia: 3358Italy: 11947
...
Munich: 13306 Trieste: 13273 Budapest: 12645 Serbia: 11590 Silesia: 6078 Rumania: 6020
Tyrolia: 5703 Vienna 5576 Galicia: 3358 Bohemia: 3248 Venice: 3160 Prussia: 3160 ...</td></tr>
</table>

Figure 6: An example of a local-agent of type Front in Spring 1902 when *Diplomat* played the role of Russia. Its type "Start" indicates that this is a new Front. Italy and Russia are the allies of the Front (*Allies*) and the enemies are Austria and Germany (*Enemies*). The *Friendly flag*, which is nil, indicates not to give weight to Italy's expected utility in choosing an appropriate strategy. The fleet in Tunis is the only unit of the other ally (Italy) that cannot be used in this Front. The local storage keeps strategies for the allies of the Front and for its enemies, a list of spaces that are important for the allies, and their values for the different allies. The three numbers that appear in the first line of the *Expected outcome*'s field of strategies are the average (Aver), minimal (Min) and maximal (Max) common expected expected profits (Aver) of the strategy. The average expected profits for each one of the allies (i.e., Russia and Italy) appears in the last line. The same list is kept for the enemies. We incorporated here only part of the actual data.

Id	10
Priority	25
Status	Run
Type	Desk
Parameters	*Power for negotiations:* Italy *Common enemies:* Austria and Germany *Type:* Check-again *Situation:* Keep *Strategy:* A Venice (I) moves to Trieste in order to attack Trieste A Vienna (R) supports A Venice to Trieste in order to attack Trieste A Bohemia (I) moves to Munich in order to attack Munich A Tyrolia(I) supports A Bohemia to Munich in order to attack Munich A Rumania (R) supports A Warsaw to Galicia in order to attack Galicia A Warsaw (R) moves to Galicia in order to attack Galicia F Sevastopol (R) supports A Rumania Expected outcomes: Aver:10617 Min: 5002 Max: 20862 Russia: 3358 Italy: 18117 *Details:* nil
Related local-agents	6
Local Storage	MESSAGE 30 FROM Russia TO Italy : I would like to suggest to you a Cooperation Agreement between Russia and Italy against Austria and Germany now END OF MESSAGE. [Stage 1 of the negotiation] MESSAGE 8 FROM Italy TO Russia : Yes,absolutely. Could you be more specific? END OF MESSAGE.

Figure 7: Desk for communications with Italy against Austria and Germany (*Common enemies*) in the middle of the negotiation period of Spring 1902. *Diplomat* played Russia. Italy and Russia tried earlier to negotiate a cooperation agreement but without any success. (*Type:* Check-again). *Diplomat* intended to keep its promises (*Situation:* keep). The strategy (*Strategy*) is the basis for the negotiations and was found by the corresponding Front. The messages in the local storage are the messages that were exchanged until now. They are actually saved in a “lisp” representation.

The only ministry that has a local agent that serves as its head is the Ministry of Defense. As we explained above, the Head-of-Defense creates the different department (Fronts) and allocates *Diplomat*'s units among them. These tasks are specific to the Ministry of Defense: Foreign Office Desks are created by Fronts and Desks are never competing for central Foreign Office resources.

3.3.2 The Ministry of Defense's local-agents

We will now demonstrate why there is a need for the creation of so many local-agents in environments similar to Diplomacy. The following reasons motivated our choice of a dynamic set of local-agents.

The Ministry of Defense is responsible for the planning and analysis. The strategies for cooperation between *Diplomat* and different powers can vary and may even contradict one another (if *Diplomat* negotiates simultaneously with hostile powers); they also change during the negotiations period. Therefore, it is not worthwhile for *Diplomat* to search for a detailed strategy for the *whole* board, but rather to search for possible *Fronts* and to find separate detailed plans for each of the Fronts. Following this method, *Diplomat* can evaluate a specific plan for a specific Front, compare it with a plan for a Front that may lead to conflicting coalitions, i.e., the set of allies of one Front intersects with the set of enemies of the other Front. Then it can decide whether to open negotiations with the allied powers of the first Front, and ultimately decide whether to sign an agreement. *Diplomat* can also change its plan for a specific Front, without changing the plans for the other Fronts and without disturbing the negotiations with other powers. In addition, *Diplomat* can save computation time by not recomputing strategies for the entire board each time it has to change the plan for a specific Front as a result of negotiations.

The need to deal with contradictory strategies is common while negotiating in a multi-agent environment, and the idea that we develop in this section may be used for other *Negotiating Automated Agent*s. These Fronts are used when deciding with whom to negotiate. We will show in Section 4 that *Diplomat* did well in this task, and we will, therefore, provide a detailed description of this algorithm.

The Head-of-Defense's local-agent searches for the possible Fronts for *Diplomat* in the beginning of each season, according to the current board game situation. It establishes a *Front* local-agent to take care of each Front it finds.

[13] All the examples are taken from Spring 1902 of a game where *Diplomat* played the role of Russia.

The easiest way to find Fronts is to consider all the combinations of the other powers as friends, enemies, or neutral powers. This method is not practical, because it can yield up to 3^6 possible Fronts. Each of these Fronts will need to find a strategy for cooperation between the allies of the Front. Therefore, the player must try to limit the number of Fronts it considers.

The need to reconsider and update the set of Fronts at the beginning of each season is as a result of the changes on the board. New estimations must be done with regards to the other agents and their relationships with one another in light of the season's orders.

The algorithm used by the Head-of-Defense to find the appropriate fronts is presented below; this algorithm was designed by trial and error. We presented to *Diplomat* situations that were taken from expert human games, let *Diplomat* provide the set of Fronts to be considered, compared it with the suggestions of human Diplomacy experts and updated the algorithm.

The algorithm extensively uses *Diplomat*'s Knowledge and Belief Base (KBB) which was described above and in particular the agreements table. The agreements table includes an entry for any agreement reached by *Diplomat* and another power. It includes the set of allies and the set of enemies of the agreement and the details of the promises for mutual cooperation (if there are such details). It also includes *Diplomat*'s intention to keep the agreement and *Diplomat*'s estimation of the intentions of its allies to keep the agreement. The intentions parameters are on a scale between 0 and 10, where 0 indicates that there is no intention to keep the agreement.

The Head-of-Defense limits the number of Fronts according to the following method:

Algorithm 3.1 Creating Fronts at the Beginning of a Season

1. *For every agreement in the agreements table do the following:*
   (a) *Establish a Front of Diplomat and the allies of the agreement against the enemies of the agreement (if such exist).*
   (b) *Establish a Front of Diplomat and the allies of the agreement against a new set of common enemies.*[14]
   (c) *Establish a Front for the set of enemies of the current agreement and Diplomat as the allies against the allies of the agreement (without Diplomat). The purpose*

[14]Possible common enemies are powers that according to *Diplomat*'s beliefs are enemies of *Diplomat* and are not friends of *Diplomat*'s allies and vice versa. For it to be worthwhile for *Diplomat* and its allies to act against these common enemies, they have to be neighbors of *Diplomat* or its allies.

of this Front is to consider reversal of alliances.

2. *For every power described in Diplomat's KBB as friendly, if the power does not already belong to the allies of the previous established Fronts, create a Front for Diplomat and the power against a set of possible new enemies.*

3. *For every power which is Diplomat's neighbor*[15] *but does not belong to the enemies set of any agreement in Diplomat's agreements table and does not already belong to the set of allies of one of the previous established Fronts, create a Front for Diplomat and the power against a set of possible new enemies.*

4. *In cases of different Fronts with exactly the same enemies, the Head-of-Defense checks whether they can be joined together, i.e. whether it will be possible to arrive at an agreement with the possible allies of the two Fronts agreeing on the basics of the same plan.*

∎

In step 1 of algorithm 3.1, *Diplomat* considers the continuation of existing agreements. The examination of possible changes in the agreement (step 1b) makes *Diplomat*'s behavior more dynamic. In step 2, *Diplomat* searches among the powers that it believes to be its friends. In step 3, it also considers its neighbors that are not believed to be enemies.

If *Diplomat* receives a message during the negotiation period announcing the existence of some agreements or offering him to establish a new coalition, it will create appropriate Fronts to check the validity of the information and when appropriate will consider the new coalition. More details on the typical behavior of the Fronts and the internal-messages they exchange with other local-agents can be found in [Kraus and Lehmann 1989].

The rules of the game at the start of a game of Diplomacy (Spring 1901) are the same as for other seasons. However there are several differences between Spring 1901 and other seasons: *Diplomat* doesn't have any information about the other powers and their relationships and doesn't have any agreements with other powers. In addition, in the Spring 1901, the units of the powers are far from each other (except Venice and Trieste). For these reasons it is very difficult to create Fronts and to decide with whom to negotiate using the algorithms used by the Head-of-Defense and the Fronts for other seasons. Therefore, we defined a special local-agent of type *Open* that is created for the first season. This local-agent uses a library

[15] Two powers are neighbors if a unit of one of them is adjacent to a unit of the other power or to a supply center which is controlled by the other power.

of openings (like openings in Chess). This library includes several openings for each power. Each opening includes information on the Fronts that should be created and possible moves to be executed in each of the Fronts, locations that should be discussed with the allies of the Front and future plans for this Front. The local agent of type Open “flips” a coin with probability that appears in the library, to choose among the possible openings.

3.3.3 The Foreign Office

The Foreign Office is responsible for the negotiation of *Diplomat* with the other powers. Since negotiation is the most important mission of *Diplomat*, we will describe the Desk local-agent in detail. A local-agent of type Desk is responsible for negotiations with a specific power. The parameters of the local-agent of type Desk are as follows (see an example in Figure 7):

1. The power for which the desk is responsible and negotiates with: its “party”.
2. The common enemies against whom it is possible to arrive at an agreement with the party (possibly an empty field).
3. The type of the negotiations may be one of the following:

   Start: Opening of negotiations.

   Continue: There exists a previous agreement with the party.

   Change enemies: There exists a previous agreement with the party. The Desk tries to change the agreement and to extend or limit the set of enemies.

   Check again: *Diplomat* and the party tried earlier to negotiate a cooperation agreement but without any success.

   Other start: The other party has started the negotiations.

   Info: Negotiations only to exchange information.

   The type of the negotiation directs the Desk local-agent's behavior during the negotiation. For example, in negotiation of type *start* the Desk will open the negotiation with a general message, suggesting an agreement. In negotiation of type *continue*, it will begin immediately with detailed discussions of possible continuation of the previous cooperation.

   Desk of type *Change enemies* is in the middle between *Start* and *Continue*: there is no need to negotiate on the formation of a new cooperation agreements as when the type is *Start*. However, the Desk can't begin in detailed discussion until it convinces the party to extend or reduce the set of enemies.

If the Desk's party starts the negotiation (type *Other start*) then the Desk evaluates its suggestion, determines the state of the negotiation (see discussion below) and continues the negotiation from that stage. In addition, if the party starts the negotiation, it will strengthen the Desk's belief that the party will keep its promises.

If the Desk and the party tried earlier to negotiate a cooperation agreement and failed (i.e., type *Check again*), then the Desk won't try to search for alternative strategies or put a lot of effort into the negotiation, since the possibility of reaching a valid agreement is lower than in other cases.

If the negotiation is of type *Info*, the Desk will send information about other powers and their relationships, or about its own plans or will require such information from its party. However, it won't give suggestions or get involved in detailed negotiation.

4. The status of the negotiations:

   (a) Arrive at a valid agreement with the intention to *keep* it.

   (b) Betrayal — Pretend to keep the agreement.

   (c) Cancel the agreement — let the partner know that you don't intend to keep the agreement.

   This parameter also influences the Desk behavior. For example, if the negotiation is in the "Betrayal" mode then the Desk will be willing to accept almost any offer from its opponent, although it will not keep it later.

5. The strategy which will be the basis for the negotiations (may be empty).

6. Details of a previous valid agreement with the other party and Diplomat (if it exists).

7. A flag that indicates whether to be friendly to the allies. This flag is used while evaluating offers. If the flag is on, the Desk will accept offers with less benefit for *Diplomat* than if the flag is off.

8. The last message sent by the other party. The message is kept for responding to it.

In the private storage area of the Desk local-agent, messages that were exchanged with the other party in the same diplomacy period are kept. Attached to every message sent by *Diplomat* the following are noted: the stage of the negotiations (see specification below) and the locations that are under discussion (if any) with the party. This information is used when the party responds.

Local-agents of type Desk are created in the following cases:

1. When a local-agent of type Front finds a good plan for cooperation between *Diplomat* and another power.

2. After receiving a message from another power with which *Diplomat* has not yet negotiated during the current season.

3. When another Desk receives a message containing information that it is worthwhile to convey to another power.

The algorithms that are used by the Desk local-agents can be found in [Kraus *et al.* 1991] Negotiation may take a long time, but usually it is a sequential process where the discussion proceeds from the general to the specifics. Accordingly, we divided the negotiation into the following stages:

1. Exchanging information about relations between powers, deciding what kind of agreement to try to achieve (cooperation or non-aggression agreement), and/or finding possible enemies or the direction of possible common attacks if they exist.

2. Negotiating about the general purposes of an agreement: spaces on the board to attack, spaces to defend, areas to leave or to enter.

3. Deciding on the specific movements in order to achieve the purposes from stage 2: which unit must attack or give support, which fleet can convoy, and which unit must move in order to cut support.

4. Signing the final agreement.

Negotiations between a Desk and its party can be in any one of those stages as is demonstrated in the example of Section 3.4. The Desk tries to run the negotiations according to stages defined above. However, human players sometimes skip a stage, or mix concepts from different stages. In such situations, the Desk recognizes the appropriate stage and continues the negotiation, as demonstrated in the example in Section 3.4.4.

3.3.4 Intelligence's Local Agents

There are three types of local-agents that belongs to the Intelligence module: Analyze-Moves, Analyze-Messages and Analyze-Builds-Retreats. The task of all three is to gather information about the other players.

The Analyze-Moves local agent is created at the end of every season. Analyze-Builds-Retreats is created at the end of Fall seasons when adjustments in strength of the players are made and builds and disbands occur. Analyze-Builds-Retreats is also created at the end of Spring seasons when there are retreats or disbands. They report their finding to the Prime Minister and the Prime Minister updates *Diplomat*'s KBB.

The detailed algorithm that they use can be found in [Kraus and Lehmann 1991]. When Analyze-Moves estimates the other players' relationships, it weights the facts to decide whether to believe two powers are friends or enemies. In general if one power gave support to another power or convoyed the other power's units it weighs in favor of them being friends. If one power attacked or helped to attack another power it weighs in favor of them being enemies. If there is a possibility that the powers planned this attack together (e.g., the unit that was attacked left its province) it weakens the assumption that the powers are enemies. Also, *Diplomat* distinguishes between the active power (the attacker or the one that gave support) and the passive one. The active one demonstrates its attitude toward the other power more precisely.

If according to its algorithm Analyze-Moves finds that there is equivalent weight in believing that the powers are friends and that the powers are enemies, Analyze-Moves makes its decision according to its old beliefs (if such exist).

Analyze-Moves also tries to deduce the other players' attitudes regarding risk from their moves. The idea behind the algorithm (see [Kraus 1988]) is to compute the difference between the most conservative strategy Analyze-Moves can think of, given its belief about the power's relationships with other powers, and the actual actions the player performed, and to bring it to a unified scale. In order to find a conservative strategy Analyze-Moves consults the Strategies Finder or local-agents of type Front.

Analyze-Builds-Retreats uses a data base that includes information about the purposes of possible builds or possible disbands of a unit (e.g., if Russia builds an army in Sevastopol, it is a possible threat to Austria or whoever controls her supply centers and building a fleet in St. Petersburg North shore is a threat against England or whoever controls the North Sea).

The local-agent of type Analyze-Messages is created at the beginning of each session. It becomes active when a declarative message is accepted by *Diplomat*. Such a message may announce that two powers are friends or enemies. We demonstrate the algorithm used by Analyze-Messages in the case *Diplomat* receives a message that announces that two powers are friends. For example:

<MESSAGE 1 FROM Turkey TO Russia> :
I would like you to know that there is a Cooperation Agreement between
Austria and Germany against Russia from the beginning.
END OF MESSAGE.

Analyze-Messages checks whether these powers appear as friends in its KBB. If so, this message just strengthens its belief that those powers are friends. Analyze-Messages announces to the Prime Minister that it should associate the name of the power that sends the message with its belief that they are friends. If later *Diplomat* verifies or refutes the existence of this relation by other sources, *Diplomat* updates the informer's loyalty measure.

If its KBB does not contain the information that these two powers are friends, it tries to estimate whether they are friends. It checks whether it is profitable for these powers to be friends and it computes the expected possible profits for those powers from signing an agreement. This is done by creating a Front local-agent with the supposed friendly powers as friends. If according to this algorithm Analyze-Messages concludes that these powers are friends, it asks the Prime Minister to update *Diplomat*'s KBB and to associate the name of the power that sent the message with this relation. Analyze-Messages uses a similar algorithm when it receives a message saying that two powers are enemies.

3.4 Example

The following is a detailed example of the internal structure of *Diplomat*, during a specific Diplomacy period. It was Spring 1902. *Diplomat* played Russia. The situation of the game is described in Figure 2. *Diplomat*'s KBB at the beginning of the season appears in Figure 4 and Figure 5.

3.4.1 Step 1

At the beginning of the season, the local-agents of type Front that appear in Figure 8 were created by the local-agent of type Head-of-Defense using Algorithm 3.1. Their private storage

areas were left empty.
The purpose of establishing Front 1 with England and Front 5 against England without the involvement of other powers (Figure 8) was to check whether to continue the non-aggression agreement with England.
The purpose of establishing Front 2 with Germany against Italy was to check the extension of the agreement with Germany. The original agreement was a non-aggression agreement.

The purpose of opening Front 3 with Turkey against Austria (Figure 8) was to check whether to continue the cooperation with Turkey.
The purpose of opening Front 4, with Austria against Italy and Turkey, was to check the possibility of betraying Turkey and starting cooperation with Austria against Turkey and Italy.
The purpose of starting Front 6 with Italy against Austria and Germany was to check whether it was worthwhile to cooperate with Italy. One can note that Italy appeared in *Diplomat*'s KBB as an enemy. Nevertheless, *Diplomat* tried to check the possibility of cooperation with her.

For each of the Fronts that the Head-of-Defense created, it writes on the notice-board the following information: the allies and the enemies of the Front, the units that it can use, the other Fronts that the coalitions they assume can exist together with the given Front's coalition (i.e., the information in Figure 9 without the "Profits" information). This information will be used later by the Fronts to compare between different coalitions.

3.4.2 Step 2

When each of the Fronts asked the help of the Strategies Finder, several possible strategies were provided for each of them. Each Front evaluated the strategies according to the Prime-Minister's personality, chose the best one and wrote the expected profits (losses) from the chosen strategy on the notice board. The information that was received from the Strategies Finder and the chosen strategy were kept in the private storage area of each Front. The new notice board appears in Figure 9.

3.4.3 Step 3

Each one of the Fronts computed the intention of *Diplomat* to keep the agreement that would be signed on the basis of the strategy found. The computation was done using the expected

Id	Priority	Status	Parameters	
1	20	Run	Front	*Allies:* England and Russia *Enemies:* — *Type:* Continue *Details:* Not enter to Barents Sea and St. Pet. *Friendly flag:* nil *Units not to use:* A Yorkshire F London
2	10	Run	Front	*Allies:* Germany and Russia *Enemies:* Italy *Type:* Change Enemies *Details:* Not enter Baltic Sea, Silesia, Prussia, Bohemia *Friendly flag:* nil *Units not to use:* A Holland
3	20	Run	Front	*Allies:* Turkey and Russia *Enemies:* Austria *Type:* Continue *Details:* Not enter Black Sea, Armenia *Friendly flag:* nil *Units not to use:* F Smyrna
4	10	Run	Front	*Allies:* Austria and Russia *Enemies:* Italy and Turkey *Type:* Check stab *Friendly flag:* nil
5	10	Run	Front	*Allies:* Russia *Enemies:* England *Type:* Check stab *Friendly flag:* nil
6	10	Run	Front	*Allies:* Italy and Russia *Enemies:* Austria and Germany *Type:* Start *Friendly flag:* nil *Units not to use:* F Tunis

Figure 8: local-agents at the beginning of Spring 1902

Id	Allies	Enemies	Units	Profits	Not Contradict
1	Russia England		A St. Pet. A Moscow	Weighted Aver:10370 Aver: 1037 Min: 1037 Max: 1037 Russia: 1037	2,3,4,6
2	Russia Germany	Italy	F Sweden	Weighted Aver: -42294 Aver: -4869 Min: -7003 Max:1530 Russia: 110 Germany: 1519	1,3,5
3	Russia Turkey	Austria	A Warsaw A Rumania F Sevastopol A Vienna	Weighted Aver: 272580 Aver:27265 Min:27045 Max:27560 Russia: 15860 Turkey: 11700	1,2,5
4	Russia Austria	Italy Turkey	A Warsaw A Rumania F Sevastopol A Vienna	Weighted Aver:9775 Aver:1188 Min: 441 Max: 1256 Russia: 595 Austria: 747	1,5
5	Russia	England	A St. Pet. A Moscow	Weighted Aver:-22301 Aver:-1990 Min:-4021 Max:0 Russia: 0	2,3,4,6
6	Russia Italy	Germany Austria	A Warsaw A Rumania F Sevastopol A Vienna	Weighted Aver:109815 Aver:10617 Min:5002 Max:20862 Russia: 3358 Italy: 18117	1,5

Figure 9: Notice board at the end of step 2. In the Profits field appear the weighted average of the common expected profits, the average of the common expected profits, the minimal of the common expected profit and maximal expected profits. On the last line, the average expected profits for each one of the allies appears. Two Fronts contradict if they assume coalitions that can't exist together. This information is kept in the "Not Contradict" column.

profits from the other Fronts.[16] As a result of this step it was decided to open negotiations with Turkey, England and Italy, and create Desks for negotiations with those powers. These parties were chosen mainly because it seems that *Diplomat* (i.e., Russia) can benefit more from forming coalitions with them than with other powers. For example, the expected profit for Russia from a coalition with Turkey against Austria was 15860 (see Figure 9), but the expected profit from forming a coalition *with* Austria against Turkey was only 595.

In addition, a new Front (7) was created to check whether it was worthwhile to continue the non-aggression agreement with Germany. The state of the other Fronts (from Figure 8) was changed to *Sleep*.

3.4.4 Step 4

To shorten the discussion, we will concentrate on the negotiations between Russia and Turkey. The Desk that was created to deal with the negotiations with Turkey appears in Figure 10.

Turkey's Desk (Desk 7 in Figure 10) wanted to start negotiations with Turkey. But before it had the opportunity to send Turkey its suggestion, *Diplomat* received the following message from Turkey:

<MESSAGE 1 FROM Turkey TO Russia> :
I would like to suggest to you the following facts:
Russia and Turkey will attack Austria now,
and Russia will attack Budapest now,
and Turkey will attack Serbia now.
END OF MESSAGE.

This message, in its internal representation, was moved to Turkey's Desk (7) by the controller. The Desk analyzed the message (see [Kraus *et al.* 1989]). The suggestion to attack Austria fits into its plans because Austria appeared in the Desk's enemies set. It was left to the Desk to check other details in Turkey's suggestion. It sent an internal-message to the corresponding Front with the following query[17]:

[16] As we mentioned before, the detailed algorithms that are used for solving different problems arising during the negotiation can be found in [Kraus *et al.* 1991]. We will explain here some of *Diplomat*'s decisions in the example, to make it complete. More details on the way the Fronts find partners for negotiation can be found in Algorithm 5.1 of [Kraus *et al.* 1991].

[17] The message was sent in an internal representation.

Id	7
Priority	25
Status	Run
Type	Desk
Parameters	*Power for negotiations:* Turkey *Common enemies:* Austria *Type:* Continue *Situation:* Keep *Strategy:* A Vienna (R) moves to Budapest in order to attack Budapest A Rumania (R) supports A Vienna to Budapest in order to attack Budapest A Bulgaria (T) supports Greece to Serbia in order to attack Serbia A Greece (T) moves Serbia in order to attack Serbia F Ankara (T) holds in order to defend Ankara A Constantinople (T) supports Bulgaria in order to defend Bulgaria A Warsaw (R) moves to Galicia in order to attack Galicia) F Sevastopol (R) supports Rumania in order to defend Rumania Expected outcomes: Aver: 27265 Max: 27045 Min: 27560 Russia: 15860 Turkey: 11700 *Details:* Not enter the Black Sea and Armenia
Related local-agents	3

Figure 10: Desk for communications with Turkey against Austria (*Common enemies*). Previous agreement between *Diplomat* and Turkey existed (*Type:* Continue). The details of the agreement were not to enter the Black Sea and not to enter into Armenia (*Details*). *Diplomat* intended to keep its promises (*Situation:* keep). The strategy (*Strategy*) is the basis for the negotiations and is found by the corresponding Front.

To: 3 *From:* 7
Contents: Evaluate the following plan:
Russia will attack Budapest now,
and Turkey will attack Serbia now.

Message 1 above which was received from Turkey, included concepts from stage 1 (type of agreements to try to achieve) and stage 2 (negotiations about general purposes of an agreement) (see Section 3.3.3). Turkey's Desk (7) analyzed the message, kept the message in its private storage area, and attached the number 2 to it to show that the negotiation had arrived at stage 2. After that, the Desk went to *Sleep* waiting for an answer to its query.

3.4.5 Step 5

When the local-agent 3 of type Front received the query, its status was changed to *Run*. It checked the suggestion that appeared in the query and found it to be a good one: it fits its own strategy (see Figure 10), and sent the following answer:

To: 3 *From:* 7
Contents: Accept the plan:
Russia will attack Budapest now,
and Turkey will attack Serbia now.
Fitted strategy:
A Vienna (R) moves to Budapest in order to attack Budapest
A Rumania (R) supports Vienna to Budapest in order to attack Budapest
A Bulgaria (T) supports Greece to Serbia in order to attack Serbia
A Greece (T) moves to Serbia in order to attack Serbia
F Ankara (T) holds in order to defend Ankara
A Constantinople (T) supports Bulgaria in order to defend Bulgaria
A Warsaw (R) moves to Galicia in order to attack Galicia
F Sevastopol (R) supports Rumania in order to defend Rumania
Expected outcomes: Aver: 27265 Max: 27045 Min: 27560
Russia: 15860 Turkey:11700
Spaces the other player mentioned: Budapest Serbia.

It is important to note that the fitted strategy is usually different from the Desk's strategy, and it is therefore important to be included in the answer.
Front 3 went to *Sleep*.

3.4.6 Step 6

Turkey's Desk (7) received the internal-message and was reactivated. It realized that it had to give a positive answer to Turkey and that the negotiations could move on to the next stage (stage 3).

The Desk found the details it wanted to suggest to Turkey (see Section 7 of [Kraus *et al.* 1991] for more details), and sent her the following message:

<MESSAGE 2 FROM Russia TO Turkey> :
In Answer to message #1: Yes,absolutely.
Now, my suggestion contains the following points:
Turkey will support by the army at Bulgaria ,
Turkey's attempt to move from Greece to Serbia,
and Turkey will move from Greece to Serbia,
and Russia will support by the army at Rumania,
Russia's attempt to move from Vienna to Budapest.
END OF MESSAGE.

According to Desk 7's strategy, there wasn't a need for close cooperation between Turkey and *Diplomat* in this season (i.e., no inter powers support or convey moves). Therefore, *Diplomat*'s discussed details of moves related to locations Turkey mentioned in her previous message (i.e., Budapest and Serbia) and kept silent about Ankara, Bulgaria and the other units. The purpose of the detailed discussion on Turkey's attack on Serbia was to help Turkey in her planning, however it turned out to be a mistake. Turkey agreed to *Diplomat*'s plan to attack Serbia from Greece, but eventually attacked it from Bulgaria. This reduced Turkey's credibility.

The Desk also updated the Prime Minister concerning the new data to be added to the agreements table regarding the agreement between *Diplomat* and Turkey against Austria: the attack on Serbia and Budapest. In addition, the Desk saved in its private storage the message that it sent to Turkey, noted the stage of the negotiations (3) and went to *Sleep* waiting for Turkey's answer.

3.4.7 Step 7

Turkey sent *Diplomat* the following message:

<MESSAGE 3 FROM Turkey TO Russia> :

In answer to message #2: Yes,absolutely.
END OF MESSAGE.

Turkey's Desk (7) got Turkey's message from the controller, and the status of the Turkish Desk was changed to *Run*. It updated the Prime Minister about the latest details (from message 2) that were agreed upon between Turkey and itself and sent Turkey the following message to confirm the agreement:

<MESSAGE 4 FROM Russia TO Turkey> :
In answer to message #3: Yes,absolutely.
END OF MESSAGE.

The Desk kept the messages in its private storage area, noted the stage of the negotiations (3) and went to *Sleep*.

3.4.8 Step 8

Turkey sent *Diplomat* the following message:[18]

<MESSAGE 5 FROM Turkey TO Russia> :
Where will Russia move from the coast of Sevastopol now ?
END OF MESSAGE.

Turkey's Desk (7) was reactivated as a result of this message and decided that it wanted to give an honest answer to Turkey. It checked its plans and sent Turkey the following message:

<MESSAGE 6 FROM Russia TO Turkey> :
In answer to message #5: I would like to suggest to you that Russia will hold at the coast of Sevastopol.
END OF MESSAGE.

It added this detail to the agreements table, stored the messages and went to *Sleep*.

[18] As was explained above, the negotiations are conducted using the formal negotiation language and some of the messages therefore are not written in correct English. The meaning of this message is actually: "Where will Russia's fleet that is based on the coast of Sevastopol move to?".

3.4.9 Step 9

Turkey, still worried about Russia's attack, sent *Diplomat* the following message:

<MESSAGE 7 FROM Turkey TO Russia> :

In answer to message # 6: I would like to suggest to you the following facts:
Russia will not move from the coast of Sevastopol
to the Black Sea now ,
and Turkey will not move from Ankara's coast to the Black Sea now.

END OF MESSAGE.

Turkey's Desk (7) sent a query concerning the message to the Front with Turkey against Austria (i.e., Front 3). We have skipped the steps that are connected with this query, and in short, Front 3 answered that it is reasonable to agree to Turkey's request not to enter the Black Sea from Sevastopol and Ankara. Therefore, Turkey's Desk (7) sent Turkey the following message:

<MESSAGE 8 FROM Russia TO Turkey> :

In answer to message #7: Yes,absolutely.

END OF MESSAGE.

Turkey's Desk (7) also added the details to the agreements table.

3.4.10 End of the session

During the negotiation period *Diplomat* didn't receive any informative messages, and the local-agent of type Analyze-Messages wasn't activated. Therefore, *Diplomat*'s beliefs about the other agents weren't changed and were as appeared in Figure 5.

However, the agreements table was changed during the negotiation. In the Sections above we have discussed only selected steps in the negotiations between Turkey and *Diplomat*. Those steps did not occur consecutively in the game, but rather other steps with regards to the negotiations between *Diplomat* and three other powers occurred intermittently. Here we described less than half of *Diplomat*'s activities in the diplomacy period. During the negotiations with Italy, the possibility of canceling the agreement with Turkey was considered, but was eventually rejected. This extensive negotiation was reflected in *Diplomat*'s agreements table at the end of the negotiation period (Figure 11). If we compare it with the agreements table at the beginning of the season (Figure 4) we can notice the following changes:

Agreement	Allies	Enemies	Intention of *Diplomat*	Intention of the Ally	Details
1	Russia England		6	7	England won't enter into Barents Sea Russia won't enter into Norway.
2	Russia Germany		4	4	Russia and Germany won't enter into Baltic Sea, Silesia, Bohemia and Prussia
3	Russia Turkey	Austria	5	5	Russia and Turkey won't enter into Armenia, Black Sea A Rumania supports A Vienna to Budapest A Bulgaria supports A Greece to Serbia F Ankara and F Sevastopol won't enter into Black Sea
4	Russia Italy	Austria	0	4	A Vienna supports A Venice to Trieste

Figure 11: *Diplomat*'s agreements table at the end of the negotiation period of Spring 1902. The intention of *Diplomat* and its allies are on a scale between 0 and 10.

Agreement 1 : the non-aggression agreement between Russia and England was strengthen (i.e., *Diplomat*'s intentions to keep the agreement increased from 4 to 6, and its estimation of England's intention to keep the agreement increased from 4 to 7). Also the details of the agreement were changed (i.e., Russia promised England not to enter Norway).

Agreement 3 : The details of the Russia-Turkey cooperation agreement (as were described above) were added to the table.

Agreement 4 : A new cooperation agreement between Russia and Italy against Austria was added to the table. Note, that *Diplomat* didn't intend to keep this agreement.

When the negotiation period ended a local-agent of type Write-Orders was created. It called the SF to find possible strategies to determine *Diplomat*'s orders. Based on *Diplomat*'s agreements table, it asked for strategies in which Turkey, England and Germany are allies and Austria and Italy are enemies. Austria was considered an enemy based on the agreement with Turkey. Italy was considered an enemy since its units bordered with *Diplomat*'s units, and *Diplomat* didn't reach a valid agreement with it. The required strategy should include the moves of Turkey-Russia agreement. In addition it specified to the SF the location that

Russia and its allies promised not to enter to according to the Turkey-Russia agreement, England-Russia agreement and Germany-Russia agreement. The Write-Orders local-agent also provided the SF with the allies' units that it wasn't allowed to use. The SF returned a set of possible strategies. The local-agent of type Write-Orders evaluated the strategies according to the Prime-Minister's personality and chose the best one. It determined what should be *Diplomat*'s moves, transferred it to the appropriate orders format and sent it to the manager of the game. The orders were:

1. A Warsaw moves to Galicia
2. A Moscow moves to St. Petersburg
3. A St. Petersburg moves to Finland
4. A Vienna moves to Budapest[19]
5. A Rumania supports A Vienna to Budapest
6. F Sevastopol supports A Rumania

Some of the orders of the other powers that were given in this session appear in the beginning of Example 3.1. Russia's orders are listed above. Other relevant orders are:

Turkey :

A Greece supports A Bulgaria to Serbia
A Bulgaria moves to Serbia
A Constantinople moves to Bulgaria
F Smyrna moves to Aegean Sea
F Ankara moves Constantinople

France :

A. Marseilles moves to Burgundy
A. Paris moves to Picardy
A. Burgundy moves to Ruhr

[19] *Diplomat*'s decision to move from Vienna to Budapest with Rumania's support was a very good move. It couldn't hold in Vienna under Italy's attack (Rumania can't support an army in Vienna and there wasn't any other Russia unit there), so it would have lost Vienna one way or the other. This way, *Diplomat* at least got Budapest and in the next season could try to get control of Vienna again. To be able to get control of Vienna in the Fall it moved the army in Warsaw to Galicia.

A. Spain moves to Gascony

A. Portugal moves to Mid-Atlantic Ocean

After the manager of the game distributed the orders of all the players and their results, a local-agent of type Analyze-Moves was created. Its task was to try to estimate the players' relationships. *Diplomat*'s beliefs before the moves can be found in Figure 5. From analyzing the moves, Analyze-Moves realized that Russia, Italy and Turkey attacked Austria (i.e., Russia attacked Budapest, Turkey attacked Serbia and Italy attacked Trieste, see the moves in this Section and in Example 3.1) and therefore Analyze-Moves concluded that they are Austria's enemies. They were added to Austria's enemies set. These beliefs were strengthened by Austria's attacks on Italy (i.e., Austria attacked Venice) and on Russia (the attack on Vienna). Italy also attacked Russia (i.e., the attack on Vienna) and Russia was kept in Italy's enemies set. Austria attacked Italy (and Italy attacked Austria) and Austria was added to Italy's enemies set. Since Turkey attacked Austria, Austria was added to Turkey's enemies set. From our insights on the game (i.e., observing the messages that were exchanged that *Diplomat* couldn't observe) it seems that Analyze-Moves conclusions were correct. The other beliefs on the players' relationships weren't changed and the updated beliefs can be found in Figure 12.

Analyze-Moves also checked which of the players kept their promises from the agreements table and reported that Italy and Turkey's credibility was decreased to the Prime Minister. That is since Italy attacked Russia even though they had an agreement, and Turkey attacked Serbia from Bulgaria and not from Greece as it promised. As we explained in Section 3.4.6 this mistrust had arisen as the result of *Diplomat*'s detailed discussion on Turkey's attack on Serbia. This discussion wasn't really necessary since there wasn't a need for close cooperation between Turkey and *Diplomat* in that attack.

Analyze-Moves also estimated the risk attitude of the other players. This was done by finding a conservative strategy for each of the players that fitted *Diplomat*'s beliefs about that player relationship with the other powers, and comparing it with that player actual moves.

In order to find a conservative strategy for Turkey, Analyze-Moves applied to the existing local-agent number 3 of type Front. This Front was for Russia and Turkey against Austria, which fitted *Diplomat*'s beliefs on Turkey's friends and enemies. Also, to find a conservative strategy for Germany it used the existing Front (7) of Germany and Russia (with no ene-

Power	Friends	Enemies
Russia	Turkey England Germany	Austria Italy
Austria		Italy Turkey Russia
England	Russia	France
France		England
Germany	Russia	
Italy		Austria Russia
Turkey	Russia	Austria

Figure 12: *Diplomat*'s beliefs about the relations between the powers at the end of Spring of 1902

mies). However, none of the other Fronts were useful to find conservative strategies for the other players and Analyze-Moves asked the Strategies-Finder to provide it with conservative strategies for the four remaining powers according to its beliefs on the players' relationships (e.g., a strategy for France against England). Based on its comparison between the conservative strategies it received from the Fronts and the actual moves of the agents it concluded that Austria is risk prone, France is risk averse and Turkey risk neutral. For example, its decision concerning Austria attitudes toward risk seems reasonable. Austria was in a difficult situation. A conservative plan would be to try to keep control over her current location and to let the units support one another (taking into consideration the possibility of cutting support). Austria, however, took a more risky plan, by trying to enter into Vienna, Trieste and Venice.

Diplomat didn't need to retreat from any location and since it was a Spring session, there was no need in deciding on builds and therefore a Retreats-Builds local agent was not created. Austria was the only power that had to retreats or disband: it had to disband its unit from Budapest and to disband its army in Serbia or to retreat with this army to Albania. Austria, of course, decided to retreat from Serbia to Albania. A local agent of type Analyze-Builds-Retreats was created. However, it couldn't conclude anything on Austria's intentions, since she actually didn't have any other choice.

3.4.11 Summary of the example

In the game from which this example was taken, *Diplomat* won the game. In general, this example demonstrates the idea of using interacting local-agents during the negotiations. From the data presented, one can see that the independence of the individual local-agents allows *Diplomat*'s negotiations with another power (in the example of Turkey) to be motivated by increasing *Diplomat*'s gains in the Front in which this power participates without fully giving up the ability to seek gains in other Fronts. This retention of the ability to consider other Fronts stems from the continued communication with other local-agents.

4 *Diplomat*'s behavior

As mentioned above, the game of Diplomacy presents a typical complex negotiation situation. We will show that *Diplomat* is a good negotiator, and this success indicates that the same structure will be useful in the creation of more Negotiating Automated agents acting in different environments.

In general, there are several ways to evaluate whether a negotiator is performing well in an environment of intensive negotiations. The main indicators are the advantages gained through her agreements and whether she achieved her goals. In addition one can analyze the negotiator's behavior during the negotiations and determine how well she did on specific tasks.

The goal of a Diplomacy player is to win the game. In long games the player needs to gain control over the majority of the board, i.e., to control 18 supply centers. In short games, in which it is predetermined exactly how many moves will be played in the game, the winner is the player with the largest number of units on the board.

4.1 The Experimental Environment

In order to test *Diplomat* we arranged several Diplomacy games. The human players were faculty and students in the department of Computer Science at the Hebrew University. All of them had played at least two long Diplomacy games before participating in the experimental games.

The negotiation was performed by electronic mail. The negotiation messages were sent to the manager of the game who distributed them among the players.

We held three or four moves per week. That is, each negotiation period lasted at least a day. We chose this procedure for two main reasons. It was very difficult for us to find enough human players that would be willing to spend long continuous periods of time playing the game. Playing a move per day or two fitted much better into their tight schedules. In addition, we didn't have a dedicated computer for *Diplomat* when we held the experiments, but used an overloaded Vax 11/785. Since we couldn't ensure that the computer would be free for *Diplomat*'s usage, we preferred to play at most one move a day. This approach provided both *Diplomat* and the human players with enough time for negotiation.

The identities of the players were kept secret and were formally known only to the manager of the game. Human players were therefore prevented from negotiating without the manager knowing. We were provided with all the messages exchanged during the negotiation and ensured that both *Diplomat* and the human players had similar negotiation procedures.

As we mentioned in Section 2, since natural language understanding was beyond the scope of this project, and in order to provide a similar setting to the human players and *Diplomat*, the messages in the negotiation were written using a formal language we had developed. The human players used a special-purpose editor which helped them compose their messages. The editor represents the composed messages in two different styles: "English" like messages for the human players and "Lisp" like messages for *Diplomat*. It allowed the human players to compose only "syntactically correct" messages, in a very restricted syntax. The examples of messages above are presented in the "English" style language[20].

Diplomat took part in the following games:

1. A long game with 6 human players. *Diplomat*, who played the role of Russia, won this game.

2. A short game. England and Russia were played by *Diplomat* and the five other powers were played by human players. The powers did not know which powers were played by humans and which were played by *Diplomat*. Russia which was played by *Diplomat* won first place and England that was also played by *Diplomat* won second place (i.e., had the second largest number of supply centers under its control).

3. A short game with 6 human players. *Diplomat* played the role of Russia and won this game.

4. A short game where *Diplomat* played all the powers. This was only a test game.

5. We presented *Diplomat* with various sub-game situations that were taken from human games. In each of those subgames, we converted the situation into *Diplomat*'s format

[20]As we explained in Section 2 we haven't put much effort in translating the "English" style syntax into a natural language. A proper treatment of the natural language interface requires a suitable treatment of pronouns, which is a much more difficult task than we were ready to take on, since natural language was not our primary interest.

and let him play. We presented *Diplomat* with the messages that the human players sent in the real game as realistically as possible. Each such subgame lasted a season. *Diplomat* played in a reasonable way, but not always the same way the human players did.

4.2 Evaluation of *Diplomat* behavior

Our findings show that *Diplomat* played well in the games in which it participated. More games are required in order to gain statistical results about *Diplomat*'s abilities in full games. We note that in the games that were held, many players did not guess which role *Diplomat* was playing.

Since Diplomacy is a very complex game, it is very difficult to identify the features of *Diplomat* that are responsible for its success. It is probably due to its overall features and the interactions between them.

Strategic Abilities: It is clear that a preliminary requirement for being a good player is the ability to find good strategies. A detailed analysis of this aspect of *Diplomat* can be found in [Ephrati 1987]. We found out that *Diplomat*'s strategic capabilities is above the average player's capabilities.

Diplomat's **Role:** Part of *Diplomat*'s success may be attributed to the fact *Diplomat* played the role of Russia and England in the games it played against human players. According to the overall record of postal games [Walker 1984] it seems that Turkey is probably the easiest country to play and Italy is probably the most difficult. Russia is in a delicate position. She wins 2 out of 9 games which end in victory of one power, yet she is also eliminated more frequently than any other power except Austria. England's insular position provides England great security and it is rarely eliminated from the game. More games are required to test how well *Diplomat* plays in the role of other powers.

Human players: It seems that even though most human players took the games very seriously, some of them didn't put enough effort into the game. Even though we promised a prize to the winner, we felt that sometimes some of the human players didn't spend enough time on the negotiation because of personal reasons. This may be the reason for our statistical findings that we present below which indicate that *Diplomat* negotiates with other parties more than human players do. It will be interesting to check how *Diplomat* will fare against players that are more motivated (e.g., games in which the prize that is offered is high).

The negotiations were conducted using a formal negotiation language. It was shown that it is rich enough to enable the players to express their negotiation messages during the diplomacy period. However, it may be that the restricted language affected the negotiation of the human players.

As we mentioned above all the human players played at least two long Diplomacy games before participating in the experimental games. It is an open question how well *Diplomat* will play against more experienced Diplomacy players.

Negotiation Capabilities: Since *Diplomat* played in more than 100 seasons (in categories 1,2,3 and 4 above), we were able to collect enough data to examine different aspects of *Diplomat*'s negotiation behavior. We checked its behavior in deciding with whom to negotiate (which depends on its local-agents of type Front), how to evaluate suggestions and predict the ally's intention to keep an agreement (which depends on the local-agents of type Desk) and to estimate the other powers' interrelations (which is under the responsibility of the local-agents of the Intelligence, Analyze-Moves and Analyze-Builds-Retreats).

With whom to negotiate: We examined *Diplomat*'s behavior in deciding with whom to negotiate. This decision is based on the creation of Fronts (see Section 3.3.2). In general, *Diplomat* needs to consider 3^6 possible coalitions in each season. Even after restricting the number of possible allies and the number of possible enemies to 3, there are still 461 possibilities.

We checked 13 seasons in the game where *Diplomat* played the role of Russia and *won* ((2) above). It considered 74 combinations of possible coalitions in the beginning of those seasons (Algorithm 3.1). In those seasons, *Diplomat* also considered 2 coalitions as a result of a message it got from another power. On average it considered 5.8 coalitions (from 461 possibilities) in a season. In those seasons *Diplomat* (Russia) was involved in 47 negotiation processes,[21] (average 3.6 per season) where in 30 negotiation processes (63% of the cases) *Diplomat* was the initiator of the negotiation (see Figure 13).

[21] We consider a negotiation process at each season that *Diplomat* exchanged at least one message with the other party.

Player	coalitions considered	Negotiation Processes in 13 seasons	Negotiation Processes average per season	initiator out of all processes
Diplomat Russia	76	47	3.61	63 %
Diplomat England	46	35	2.69	48%
Austria	—	42	3.2	52%
France	—	27	2.07	33 %
Germany	—	23 (in 6 seasons)	3.83	60%
Italy	—	30	2.3	33%
Turkey	—	18 (in 10 seasons)	1.72	47%
Diplomat average	61	41	3.15	55%
Human average		33 without (G) and (T)	2.62	45%

Figure 13: *Diplomat*'s and other players' behavior during the negotiation period. The second column (from left) indicates how many coalitions *Diplomat* considered. No data is available about the number of possible coalitions the human players considered. The third column indicates the number of negotiation processes in which a specific player was involved. The last column is the percentage of the negotiation processes initiated by the specific player. Germany was eliminated from the game after 6 seasons. Turkey was eliminated from the game after 10 seasons.

player	messages that were checked	message was understood	
		messages	%
Diplomat	114	101	89%
Humans	90	83	92 %

Figure 14: Summary of performance while evaluating suggestions. The messages we consider as "messages that were understood" are those that the player did not ask for clarifications from the sender of the message.

In the same game ((2) above) where *Diplomat* played the role of England and finished second, it considered 46 coalitions in 13 seasons. Two coalitions were considered as a result of a message it got from another power. In this case the average is 3.5 coalitions per season. In those seasons, *Diplomat* (England) was involved in 35 negotiation processes (average 2.69 per season) where in 17 cases *Diplomat* (England) was the initiator (48%).

We compared these findings with the behavior of human players. The average number of negotiation processes for a power varied between 3.28-1.72 with the average of 2.5 negotiation processes for a power per season. On average, each power was the initiator of the negotiation process in 45% of those cases. These findings indicate that *Diplomat*, on the average, negotiates with other parties more than human players do.

Evaluation of suggestions: In order to analyze *Diplomat*'s behavior while evaluating suggestions (this is done by the Desks local-agents), we examined 114 detailed suggestions that were received by *Diplomat* during negotiation periods (see Figure 14). In 13 (11.4%) cases, *Diplomat* asked for more details from its opponent because it could not find a strategy that would enable it to evaluate the suggestion. We also examined 90 suggestions that were received by human players. We found out that in 7 (7.7%) cases they asked for more details. In this task *Diplomat* did slightly worse than human players but still succeeded in evaluating 88.6% of the suggestions we have examined.

Estimation of other powers' interrelations: We also tried to evaluate how effective *Diplomat* was in estimating the other powers' interrelations. We examined 260 relations between powers in a Diplomacy game. In 223 cases (85.7%) *Diplomat* estimated correctly the existence of those relations. In 25 cases (9.6 %) *Diplomat*

	both kept	both broke	*Diplomat* broke	other broke	sum
sum	51	4	6	2	63
Diplomat correct	51	3	4	0	58

Figure 15: *Diplomat* and its allies behavior after signing agreements. 63 agreements were checked during 23 seasons. We distinguish between 4 cases (from left): *Diplomat* and its allies kept the agreement, *Diplomat* and its allies broke the agreement, *Diplomat* broke the agreement and its allies kept it and vice versa. The number of agreements in which *Diplomat* estimated correctly the intention of its allies to keep an agreement is listed in the second row.

did not discover relations that existed and in 12 cases (4.6 %) *Diplomat* estimated that relations existed and was wrong. According to our experience, *Diplomat*'s estimations are at least as good as human players'.

Prediction of the ally's intention to keep an agreement: We also examined 63 agreements that were signed between *Diplomat* and another power, and *Diplomat* predicted successfully the ally's intention to keep an agreement in 92% of all the agreements it had signed (see Figure 15). The human players predicted successfully *Diplomat*'s intention to keep an agreement in 86% of all the agreements they had signed with *Diplomat*.

We conclude that in the tasks that we examined, *Diplomat* performed at least as well as human players.

Diplomat's **Structure:** Next, we examined the number of local-agents that were created during the short games *Diplomat* played. We have chosen a dynamic structure for *Diplomat*, where local-agents are created when they are needed for a specific task, and when the task for which the local-agent has been created is performed or is no longer relevant, the local-agent vanishes. We have also considered a static structure, where the local-agents are created at the beginning of the game and last until the end of the game. Our findings which we explain below, indicate that the static structure is not beneficial since a lot of local-agents are created, and at least half of them are essentially different from one another or are created at least one season apart. Therefore, it is not worthwhile to keep them in *Diplomat*'s memory for the whole game.

A local-agent of type Front is essentially different from another local-agent of type Front, if the Fronts' sets of allies are not equal or the Fronts' sets of enemies are not

Role & Game	# Fronts	# Desks	Sum	# seasons	Average Per-Season	Different Local-agents	Not-continuous	% of both
Russia 2	76	51	127	13	9.8	42	13	43%
England 2	46	35	81	13	6.2	31	6	46%
Russia 3	28	17	45	4	11.2	25	0	55 %

Figure 16: The local-agents of type Front and Desk that were created during two games (2 and 3 above). Two Fronts are different if either their allies sets or their enemies sets are not the same, and similarly for Desks. The number of local-agents of type Desk and Front that were different in each game appear in the seventh column. The "non-continuous" column includes the number of local-agents of type Desk or Front that were similar to previous local-agents, but were created at least one period apart from their similar ones. The last column indicates the ratio of the sum of columns 7 and 8 to column 4.

equal. This applies similarly to Desks.

Our findings indicate that in game 2, which lasted 13 seasons, in which *Diplomat* played the role of Russia, 127 local-agents of type Front and Desk were created in the 13 seasons of the game (see Figure 16). 42 out of these local-agents were essentially different. Among the rest of the local-agents, 13 were created at least one season apart from their similar ones. Therefore, it was not necessary to keep 43% of the local agents of type Front and Desk for the whole game. In addition local-agents of type Head-of-Defense, Write-Orders, Analyze-Moves, Retreats-Builds and Analyze-Builds-Retreats were created at the end of each season and were not used during the diplomacy period. Note that even if a local-agent of type Front (or Desk) is created in one season, and then a similar one is created in the next season, the other parameters (besides the allies and enemies which are the same) are usually different and the board of the game is different.Hence, even in such a situation it is not clear that any previous information of the local-agent is relevant.

Similarly, when *Diplomat* played the role of England in the same game, 82 local- agents were created, in which 46% were different or were created at least one season apart.[22]

When *Diplomat* played the role of Russia in the third game, 45 local-agents were created in 4 seasons. 55% of these local-agents were different.

[22]We note that in this game Russia won the game and England came in second. We suspect that it is related to the number of local-agents that were created. When *Diplomat* played the role of Russia, more local-agents were created, i.e., it considered more possible coalitions and negotiated with more powers which is an indication of a good negotiator.

To summarize, these statistics indicate that in a complex negotiation situation as the game Diplomacy, successful negotiation requires large number of local-agents, and that about 50% of them should be different. These findings support the idea of a dynamic local-agents set. When there are so many local-agents that are needed for only short periods of time, it is logical to create a new local-agent when it is required, rather than to keep a large set of local-agents which is space consuming. The idea of dynamic local-agents is also advantageous to the maintenance of a fixed small set of local-agents, because either the small fixed set requires time for allocating the tasks among the agents or the local-agents will waste time bidding for the tasks. The efficiency of the dynamic local-agents set method in time and space is especially important in real-time negotiations.

We think that *Diplomat*'s framework could be extended to any *Negotiating Automated Agent* acting in similar situations. We describe our ideas in the next Section.

5 General Structure for a Negotiating Automated Agent

5.1 Modules

We think that the general structure of any *Negotiating Automated Agent* could be modeled after that of *Diplomat* It should contain the same modules: the *Prime Minister* that directs the *Negotiating Automated Agent*'s activities, the *Ministry of Defense* that is responsible for the planning, the *Foreign* office that negotiates with the other agents, the *Headquarters* that executes the basic tasks of the agent, and the *Intelligence* that is responsible for collecting information about the environment and the other agents.

In each application the modules should be adjusted to the specific situation in which the *Negotiating Automated Agent* acts and negotiates. For example, in the case of a negotiation between companies, the Research and Development department of a company (or the Financial operations department) carries out the task of the ministry of Defense. The negotiation team plays the role of the Foreign office. The Chief Headquarters Officer (or the headquarters team) is responsible for the implementation of the company's activities: e.g., giving orders to the bookkeeper or to the storekeeper to carry out the details of a signed agreement. In such environments the Intelligence unit is usually hidden and is not named. It is either part of Research and Development or it is referred to as an industrial or financial intelligence unit.

5.2 Relationships Between the Modules

As we demonstrated in Diplomacy, the development of different modules for negotiation and planning is a characteristic of a good negotiator (see [Fisher and Ury 1981]). A good negotiator should do much "inventing". That is, find out new ideas that are not already among the negotiation issues. The separation of the planning and negotiation into two modules enables the ministry of Defense to find as many solutions to the problem as possible without taking into account whether they are acceptable to the other side. The ideas won't be conveyed to the other side until the Foreign Office decides to do so. Therefore, their consideration by the ministry of Defense can do no harm.

The separation between the Headquarters and Foreign Office modules is to enable flexible behavior during the negotiation period. The Foreign Office may offer different and even contradictory agreements to other agents, but when the time comes to carry out the actions, the Headquarters considers the different agreements that were reached and decides which to implement. The Headquarters can consult with the ministry of Defense in making the decision, but the ministry of Defense's task is only to provide different plans and solutions and to estimate their effectiveness. The Headquarters, however, makes the decision. Once again, the ministry of Defense can provide several solutions and plans, and the best will be chosen.

Intelligence serves a different task, that is useful to all the other modules. The information that it collects regarding relations between the parties may help the Foreign Office module during the negotiation to estimate the other agents' intentions. Information about tactical decisions of the agents may serve the ministry of Defense while evaluating strategies. It is worthwhile, therefore, to dedicate a separate module to this task.

The Prime Minister directs and influences the behavior of all the modules and maintains the *Negotiating Automated Agent* data base. It gathers the information from all the other modules and serves as a central and unified component between the modules giving them a personality. This yields the consistent overall behavior of the *Negotiating Automated Agent*.

The division into independent modules is also beneficial during the development period (as we realized during *Diplomat*'s development) because each module can be developed independently.

5.3 The Design of the Modules

The ministry of Defense is responsible for finding appropriate plans for the different tasks in the environment. One of our conclusions from the development of *Diplomat* is that in a dynamic and changing environment, it is usually not worthwhile to search for a global plan for solving all the problems and fulfilling all the tasks. Rather, it is more useful to divide each of these problems into subproblems and the ministry of Defense can look for small plans for solving them. These plans can serve as the basis for negotiations, and can be updated during the negotiation process. This is similar to updating plans in reactive planning systems (e.g., [Georgeff and Lansky 1987; Kaelbling 1987]. When the *Negotiating Automated Agent* needs to take an action, the small updated plans may be incorporated into a global plan.

Another one of our conclusions is that the division of Foreign Office module into independent delegations is very important when the *Negotiating Automated Agent* holds simultaneous negotiations with several other agents, and it becomes even more so when the *Negotiating Automated Agent* may possibly not keep its promises. In such a case the *Negotiating Automated Agent* may suggest contradictory offers to different agents, and it will need to search for contradictory strategies.

Possible problems may arise from the communications with the other agents in such an environment; therefore, there is a need for independent delegations to communicate with other agents, but for each agent, there exists only one such delegation at a given time. Therefore, it might happen that contradictory messages will be sent to different agents, but not to the same agent. The composition of each delegation, its negotiation strategies and its general behavior may be changed as a result of changes in the environment.

Intelligence is responsible for collecting information about the environment and the other agents. This task is necessary since we assume incomplete information and dis-trustful agents. It also keeps track of whether the other agents have kept their promises. One of its most important tasks is to try to predict the other agents' actions and to estimate the relationships between the agents. There may be different sources of information, and this module should also be subdivided into sections that are responsible for the different types of information collecting.

5.4 Local Agents

The idea of a dynamic set of independent local-agents can be useful in the design of any autonomous agent that acts in a complex and dynamic environment. In the case of the

Negotiating Automated Agent it seems especially useful in a case of simultaneous negotiations as we demonstrated in Diplomacy.

In general, when developing a specific *Negotiating Automated Agent* the tasks that should be performed by each of the modules must be specified. When the *Negotiating Automated Agent* is active, if a specific need for fulfilling a task arises (the task is under the responsibility of one of the agent's modules), a local-agent is created with the type that is appropriate for the specific task. The variables of the local-agent are initialized according to the specific circumstances in which it is created. When the task for which the local-agent has been created is performed or is no longer relevant, the local-agent vanishes. We described the structure of local-agents in Section 3.3.1.

A local-agent cannot directly modify another local-agent, and all the communications between the local-agents are done through the mail-box or the notice board. This strengthens the independence of each of the local-agents. The main idea behind the independence of the local-agents is to enable each of them to try to perform its mission in the most efficient way, even if it may contradict other local-agents activities. This is important if the *Negotiating Automated Agent* holds simultaneous negotiations with several other agents and it becomes even more important when the *Negotiating Automated Agent* may possibly not keep its promises. As we mentioned above, in such a case the *Negotiating Automated Agent* may suggest contradictory offers to different agents, and it will need to search for contradictory strategies. The local-agents also have to be designed in a way that the *Negotiating Automated Agent* will act consistently.

We compare our local-agents structure with similar structures in the end of the next section.

6 Related work in DAI

Research in Distributed Artificial Intelligence (DAI) is concerned with how automated agents can be designed to interact effectively (for a survey of DAI see [Bond and Gasser 1988; Gasser 1991]). The subject of negotiation, that we have discussed in this paper, is one of the subjects that has occupied the efforts of the DAI community (e.g., [Smith and Davis 1983; Georgeff 1983; Sycara 1987; Malone *et al.* 1988; Conry *et al.* 1991; Durfee 1988; Rosenschein and Genesereth 1985; Sathi and Fox 1989; Zlotkin and Rosenschein 1990; Kreifelts and Martial 1990; Durfee and Montgomery 1990].

Davis and Smith's work on the Contract Net [Smith and Davis 1983] introduced a form of simple negotiation among cooperative agents, with one agent announcing the availability of tasks and awarding them to other bidding agents. Malone refined this technique considerably by overlaying it with a more sophisticated economic model [Malone *et al.* 1988], proving optimality under certain conditions. While Davis and Smith's original work assumed some autonomy among agents, these agents willingly bid for tasks without explicit motivation. Malone's work introduced a motivational framework in the language of economic theory, and at the same time provided a more theoretical language in which to discuss the task-sharing algorithm.

These efforts in DAI and others that have followed dealt with negotiations in the case of cooperative systems which are designed to achieve a common general task, or in which the agents belong to the same organization or unit (see for example [Georgeff 1983] which describes a method for synthesizing multi-agent plans from simple single-agent plans, [Sathi *et al.* 1986] which deals with project management, [Durfee 1988; Durfee and Lesser 1989] which deals with the vehicle monitoring domain, and [Sathi and Fox 1989] which deals with resource reallocations). Conflicts among the agents in these environments may arise while each tries to achieve its own sub-tasks (for example, they may need to share the same resources), but their overall task is the same.

We study the negotiations that could occur among agents that serve the interests of truly distinct parties (i.e., in Diplomacy each player plays the role of a different power). In our environment, the agents are autonomous; they have their own utility functions (i.e., in Diplomacy, the utilities are based on the number of supply centers controlled by the player), and no global notion of utility plays a role in their design. The agents are *individually motivated.*

Other works that deal with similar situations are [Sycara 1987; Rosenschein and Genesereth 1985; Zlotkin and Rosenschein 1989; Kraus and Wilkenfeld 1991a; Kraus and Wilkenfeld 1991b]. Sycara [Sycara 1987] presented a model of negotiation that combines case-based reasoning and optimization of the multi-attribute utilities of the agents. She implemented her ideas in a computer program called the PERSUADER which resolved adversarial conflicts in the domain of labor relations, and tested her system using simulations of such domains. While she concentrated on the perspective of the mediator (see also [Kolodner and Simpson 1989]), we analyze such situations from the point of view of the participants.

Matwin et al. [Matwin *et al.* 1989] developed an expert system shell called *Negoplan* to support single party participants in a negotiation. *Negoplan* simulates the changes in the positions of the parties during the negotiation, based on their anticipated behavior.

Their method does not simulate the entire process of negotiation since they give one party a competitive advantage. In simulating the overall negotiation process, we concentrate on comparisons between one attribute subject of the negotiation and the outside options available to the negotiator.

Rosenschein and Genesereth [Rosenschein and Genesereth 1985], used certain game-theoretic techniques to model communication and promises in multi-agent interaction. In their study, the process of negotiation was severely restricted (the agents could only make single, simultaneous offers), and it assumed that each agent knew the complete payoff matrix associated with the interaction. For large games involving many agents and outcomes, the kind of environments in which we are interested, the size of a payoff matrix may quickly become intractable. This work was extended by Zlotkin and Rosenschein in [Zlotkin and Rosenschein 1989] and [Zlotkin and Rosenschein 1990; Zlotkin and Rosenschein 1991], and in these last works there is no need for the players to know the full payoff matrices, but the players still need to explore all possible strategies. Kraus and Wilkenfeld used different game theory techniques to model negotiation of automated agents under time constraints ([Kraus and Wilkenfeld 1991a; Kraus and Wilkenfeld 1991b]). They also assume that the set of possible agreements is limited and that there is full information, but they make fewer restrictions on the negotiation procedure.

Comparing this work to our work, our system acts and negotiates successfully with human partners in a more complex environment where a mediator is not available. The agents may break their promises, and possible coalitions between other agents must be taken into account. Close cooperation between different agents is needed.

We would like to compare our internal structure with the blackboard framework [Hayes-Roth and Lesser 1977; Hayes-Roth 1985; Corkill *et al.* 1987], with message-passing systems for problem solving [Gasser *et al.* 1987] and with the distributed problem solvers mechanism [Durfee and Lesser 1987].

In our system we enable our local-agents to communicate using both methods, the blackboard mechanism and message passing. The blackboard mechanism is used for message broadcast (e.g., for resource allocation), and the message passing is used for bilateral message exchange. But the major difference between our system and the others is the dynamic changes of the local-agent set. In the blackboard systems the set of knowledge sources are fixed. The other distributed problem solvers systems are also comprised of a fixed collection of problem-solving agents, each of which has a stable micro-architecture. In some systems, the set of inter-agent relationships is changed yielding a dynamically adaptive macroarchitecture. For example, in the Distributed Vehicle Monitoring Testbed (DVMT),([Durfee and Lesser 1987]), there is a fixed set of vehicle monitoring nodes, where each node is responsible for a preliminary assigned portion of the sensed area.

These methods are not applicable in complex and dynamic environments such as Diplomacy. We have developed a dynamic local-agents structure where local-agents are created when their services are needed and are destroyed when their services are no longer needed (because they have fulfilled their goal or changes in the environment have made it no longer relevant). This dispensibility saves storage space and the need for spending time in allocating agents to tasks (for example, the bidding mechanism of [Smith and Davis 1983],) and is appropriate for a *Negotiating Automated Agent* that acts in a complex changing environment like Diplomacy.

Recently, Gasser and Ishida [Gasser and Ishida 1991] have started to investigate the idea of agents which actively and flexibly construct and reconstruct themselves. They conclude that this kind of system offers the promise of a wider degree of organizational flexibility. They have concentrated on production systems.

Our findings about *Diplomat* (see Section 4) indicate that a large set of different local-agents are created during the negotiation periods, and that the dynamic local-agent structure is useful. In the next section we explain how *Diplomat*'s structure can be used for building a Negotiating Automated Agent that acts in other environments.

7 Conclusion and Future Work

We have proposed methods for building a *Negotiating Automated Agent* that participates in a complex environment where the set of possible strategies is very large, a mediator is not available, the agents may break their promises, close cooperation between different agents is needed, and possible coalitions between other agents must be taken into account. We used these methods to develop the system *Diplomat*, which plays the game of Diplomacy.

One set of open questions is how well *Diplomat* will play in other situations (e.g., better players, real-time negotiation, other roles in the game). More experiments are needed in order to check these issues. However, we found out that carrying out such experiments is quite difficult due to the long time a Diplomacy game lasts. It is also hard to identify the exact contribution of each part of *Diplomat* to its success. Another direction for future research is to try to use *Diplomat*'s techniques in a simpler environment. As a first step in this direction, we recently have developed a model of Hostage-Crisis negotiation simulation [Kraus *et al.* 1992;

Kraus and Wilkenfeld 1993]. There the number of options available to the players is much smaller than in Diplomacy. We have analyzed the Hostage Crisis model using game-theory techniques and held whole human simulations. We are in the process of constructing a Negotiating Automated Agent that will participate in Hostage Crisis simulations.

Diplomat acts in a multi-agent environment, where each agent is autonomous and self motivated. In addition, there are no well defined negotiation protocols, i.e., the structure of the negotiation in Diplomacy is not defined or restricted by the rules of the games. Other situations where the agents are autonomous and self motivated and the negotiation protocols are not structured may occur when automated agents negotiate with humans, for example, in training humans in negotiation or games. Also, such negotiating-automated agents may be useful in helping humans in negotiation with other humans. We strongly suspect that *Diplomat*'s framework, together with the methods and algorithms that we developed for *Diplomat*'s negotiation can be used in such environments after some modifications.

8 Acknowledgments

This work was partially supported by National Science Foundation under grant number IRI-9123460. It was partially done while Sarit Kraus was at the Hebrew University. We would like to thank Eitan Ephrati for developing the Strategies Finder and to Arie Schlesinger for developing the special editor for the Negotiation Language. We would also like to thank the Avalon Hill Company for allowing us to reproduce their Diplomacy map in this article.

References

Bond, A. H. and Gasser, L. 1988. An analysis of problems and research in DAI. In Bond, A. H. and Gasser, L., editors, *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., San Mateo, California. 3–35.

Conry, S.E.; ; K.Kuwabara, ; Lesser, V.R.; and Meyer, R.A. 1991. Multistage negotiation for distributed satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence* 21(6):1462–1477.

Corkill, D. D.; Gallagher, K.Q.; and Johnson, P.M. 1987. Achieving flexibility, efficiency, and generality in blackboard architectures. In *Proceedings of AAAI-87*. 18–23.

Durfee, E. H. and Lesser, V.R. 1987. Global plans to coordinate distributed problem solvers. In *Proc. of IJCAI-87*. 875–883.

Durfee, E. H. and Lesser, Victor R. 1989. Negotiating Task Decomposition and Allocation Using Partial Global Planning. In Gasser, L. and Huhns, M. N. editors, *Distributed Artificial Intelligence, Volume II*. Pitman/Morgan Kaufmann, London. 229–244.

Durfee, E. H. and Montgomery, T. A. 1990. A hierarchical protocol for coordinating multiagent behaviors. In *Proc. of AAAI-90*, Boston, MA. 86–93.

Durfee, E. H. 1988. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston.

Ephrati, E. 1987. Strategy - planning an optipmal strategy for a diplomacy player. M.Sc. thesis, Dept. of Computer Science, Hebrew University, Jerusalem (Written largely in Hebrew).

Fisher, R. and Ury, W. 1981. *Getting To Yes: Negotiating Agreement Without Giving In*. Houghton Mifflin, Boston.

Gasser, L. and Ishida, T. 1991. A dynamic organizational architecture for adaptive problem solving. In *Proc. of AAAI-91*, California. 185–190.

Gasser, L.; Braganza, C.; and Herman, N. 1987. Implementing distributed artificial intelligence sysems using mace. In *Proc. of the Third IEEE Conference on Artificial Intelligence Applications*. 315–320.

Gasser, L. 1991. Social concepts of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence* 47(1–3):107–138.

Georgeff, M. and Lansky, A. 1987. Reactive reasoning and planning. In *Proc. of AAAI-87*, Seattle, WA. 677–682.

Georgeff, M. 1983. Communication and interaction in multi-agent planning. In *Proc. of the National Conference on Artificial Intelligence*, Washington, D.C. 125–129.

Hayes-Roth, F. and Lesser, V.R. 1977. Focus of attention in the heasay-ii system. In *Proc. IJCAI-77*, Cambridge,MA. 27–35.

Hayes-Roth, B. 1985. A blackboard architecture for control. *Artificial Intelligence* 26(3):251–321.

Kaelbling, L. 1987. An architecture for intelligence reactive systems. In Georgeff, M. and Lansky, A., editors, *Reasoning about Actions and Plans*. Morgan-Kaufmann, Los Altos, California.

Karrass, C.L 1970. *The Negotiating Game: How to Get What You Want*. Thomas Crowell Company, NY.

Kolodner, J. L. and Simpson, R. L. 1989. The mediator: Analysis of an early cased-based problem solver. *Cognitive Science* 13:507–549.

Kraus, S. and Lehmann, D. 1989. Designing and building an automated negotiator. Technical Report UMIACS TR 89-105, CS TR 2338, Institute for Advanced Computer Studies, University of Maryland.

Kraus, S. and Lehmann, D. 1991. Knowledge acquisition in negotiation games. In Geyer, F., editor, *The Cybernetics of Complex Systems - Self- organization, Evolution, and Social Change*. Intersystems Publications, Salinas, CA.

Kraus, S. and Wilkenfeld, J. 1991a. The function of time in cooperative negotiations. In *Proc. of AAAI-91*, California. 179–184.

Kraus, S. and Wilkenfeld, J. 1991b. Negotiations over time in a multi agent environment: Preliminary report. In *Proc. of IJCAI-91*, Australia. 56–61.

Kraus, S. and Wilkenfeld, J. 1993. A strategic negotiations model with applications to an international crisis. *IEEE Transaction on Systems Man and Cybernetics* 23(1):313—323.

Kraus, S.; Ephrati, E.; and Lehmann, D. 1989. Evaluation of suggestions during automated negotiations. In *Proc. of the 11th Cognitive Science Conference*. 299–306.

Kraus, S.; Ephrati, E.; and Lehmann, D. 1991. Negotiation in a non-cooperative environment. *J. of Experimental and Theoretical AI* 3(4):255–282.

Kraus, S.; Wilkenfeld, J.; Harris, M.; and Blake, B. 1992. The hostage crisis simulation. *Simulations and Games* 23(4):398—416.

Kraus, S. 1988. *Planning and Communication in a Multi-Agent Environment.* Ph.D. Dissertation, Hebrew University, Jerusalem. (Written largely in Hebrew).

Kreifelts, T. and Martial, F. Von 1990. A negotiation framework for autonomous agents. In *Proc. of the Second European Workshop on Modeling Autonoumous Agents in a Multi Agent World*, France. 169–182.

Levy, J. and Dimitrovski, Y. 1984. The ylisp 2 manual. Technical report, Hebrew University, Jerusalem.

Malone, T. W.; Fikes, R. E.; Grant, K. R.; and Howard, M. T. 1988. Enterprise: A marketlike task schedule for distributed computing environments. In Huberman, B. A., editor, *The Ecology of Computation.* North Holland. 177–205.

Matwin, S.; Szpakowicz, S.; Koperczak, Z.; Kersten, G. E.; and Michalowski, W. 1989. Negoplan: An expert system shell for negotiation support. *IEEE Expert* 4(4):50–62.

Rosenschein, J. and Genesereth, M.R. 1985. Deals among rational agents. In *Proc. of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, California. 91–99.

Sathi, A. and Fox, M.S. 1989. Constraint-directed negotiation of resource reallocations. In Gasser, L. and Huhns, M. N., editors, *Distributed Artificial Intelligence, Volume II.* Pitman/Morgan Kaufmann, London. 163–194.

Sathi, A.; Morton, T.E.; and Roth, S.F 1986. Callisto: An intelligent project management system. *The AI Magazine* 7(5):34–52.

Schlesinger, A. 1987. A natural language system for the diplomacy game. M.Sc. thesis, Dept. of Computer Science, (written in Hebrew).

Smith, R.G. and Davis, R. 1983. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* 20:63–109.

Sycara, K.P. 1987. *Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods.* Ph.D. Dissertation, School of Information and Computer Science, Georgia Institute of Technology.

The Avalon Hill Game Co., 1976. *Rules for Diplomacy.*

Walker, R. 1984. *A Gamer's Guide to Diplomacy.* The Avalon Hill Game Company, Baltimore, Maryland.

Wilkins, D. 1983. Using pattern and plans in chess. *Artificial Intelligence* 14(2):165–203.

Zlotkin, G. and Rosenschein, J. S. 1989. Negotiation and task sharing among autonomous agents in cooperative domains. In *Proceedings of IJCAI-89*, Detroit, Michigan. 912–917.

Zlotkin, G. and Rosenschein, J. 1990. Negotiation and conflict resolution in non-cooperative domains. In *Proceedings of AAAI-90*, Boston,MA. 100–105.

Zlotkin, G. and Rosenschein, J. 1991. Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence* 21(6):1317–1324.

List of Figures