Cooperation and Competition in Emotional Robot Societies

Author: Pablo Gómez Esteban
Supervisor: David Ríos Insua

Statistics and Operational Research Department

Universidad Rey Juan Carlos

Ph.D. Dissertation

February 2014

Don **David Ríos Insua**, Catedrático del Departamento de Estadística e Investigación Operativa de la Universidad Rey Juan Carlos.
AUTORIZA:
La presentación de la Tesis Doctoral titulada

Cooperation and Competition in Emotional Robot Societies

Realizada por Don Pablo Gómez Esteban bajo su dirección y supervisión y, que el Departamento de Estadística e Investigación Operativa ha dado su conformidad para que sea presentada ante la Comisión de Doctorado.

En Móstoles, a 4 de Febrero de 2014.

Fdo.: D. David Ríos Insua

Don **Antonio Alonso Ayuso**, Profesor Titular de Universidad y Director del Departamento de Estadística e Investigación Operativa de la Universidad Rey Juan Carlos.
INFORMA:
Que la presente Tesis Doctoral titulada

Cooperation and Competition in Emotional Robot Societies

Ha sido realizada por Don Pablo Gómez Esteban bajo la dirección y supervisión de Don David Ríos Insua y, que el Departamento de Estadística e Investigación Operativa ha dado su conformidad para que sea presentada ante la Comisión de Doctorado.

En Móstoles, a 4 de Febrero de 2014.

Fdo.: D. Antonio Alonso Ayuso

To my family.

Acknowledgements

These two years and a half have been such an adventure: learning so much, traveling, getting to know amazing people, discovering illusions and frustrations, etc. All of that would not have been possible without the help of my supervisor, David Ríos. I would like to thank him for giving me this great opportunity, for the countless amount of hours that I, and this thesis, have stolen to him and his family, and finally, for his patience and love on his corrections. I would like to extend this acknowledgement to the AiSoy Robotics team, specially to Diego García, Pablo García and Carles Cerezo.

Then, I would like to acknowledge my parents for their inconditional support. I would not be who I am today without their complete dedication and love.

I would not have found the determination and confidence required to finish this thesis without Nuria. For being by my side and my reference point to get over the difficulties. Thank you.

As a second family, I could not forget to thank my friends for sharing dreams, support, laughs and talks. From Edinburgh to Madrid, a big thanks to all of you.

Throughout these two years, I have had the chance to meet such inspiring people that, at different levels, have helped me in completing this work. I would like to thank Michael Rovatsos, Javier Martínez Moguerza, Marta Sanz, Javier Cano, José María Lavín, César Alfaro and Javier Gómez for, probably inadvertently to them, having inspired me so many times, talking to them or even just listening to what they had to say. Apart from them, to Alberto and Martti for having shared such good moments and experiences.

Finally, I would like to extend the acknowledgement to all the Statistics and Operational Research department, especially to its director, Antonio Alonso Ayuso, for the warm welcome they gave me.

This thesis has been developed supported by grants from the MICINN project RIESGOS, the RIESGOS-CM project and the INNPACTO project HAUS.

Contents

Abstract

One of the current challenges in robotics is the development of models that may be implemented in non-expensive platforms, with its entailed computational limitations. Our motivation for this thesis is the design and implementation of affective decision-making models within low-cost domestic robots. These agents will form societies in which they interact among them and with one or more users, facilitating their adoption in application areas such as companion pets to provide emotional support to e.g. the elderly or autistic kids. The described models have been implemented within an AiSoy1 robot, a non-expensive platform, whose processing capacities lead us to develop simple, yet rigorous, models.

Through this work, we describe Adversarial Risk Analysis based behavioral models for an autonomous decision agent which processes information from its sensors, facing several intelligent adversaries using multi-attribute decision analysis at its core, complemented by models forecasting the decision-making of adversaries.

We also explore the social needs of a robotic agent and how it handles interactions in which several robotic agents compete among themselves in their interaction with several users, or cooperate to satisfy themselves and the users. As we refer to autonomous agents, they may move from a cooperative to a competitive attitude, and vice versa, depending on their previous interaction experience.

As we are expecting robots to become part of our daily lives, we would need them to be as interactive as possible, including affective mechanisms. Therefore, we describe a model that is a compound of emotions and mood, in which the agent behavior is influenced by such affective phenomena in different ways.

Resumen

El ser humano ha estado permanentemente interesado en construir máquinas que le ayuden a realizar tareas que hagan su vida más fácil. Los avances realizados en el campo de la robótica han tratado de exprimir al máximo las capacidades computacionales de las plataformas disponibles, para lograr así imitar y complementar el comportamiento del ser humano. Actualmente, uno de los principales retos en la robótica es desarrollar modelos relevantes que puedan ser implementados en plataformas tecnológicas sencillas, con sus limitaciones computacionales pero disponibles a un precio asequible.

Tradicionalmente las emociones han sido consideradas como un factor limitante en el proceso de toma de decisiones. Sin embargo, recientes estudios en el campo de la neurología, demuestran que actuar sin capacidad emocional puede resultar muy complicado o incluso imposible en ciertas ocasiones. Estos resultados se han complementado con trabajos psicológicos que demuestran la influencia que los factores emocionales, o afectivos, tienen sobre nuestros procesos cognitivos.

El interés por incluir dichos factores afectivos en máquinas ha cobrado especial relevancia con la aparición del campo de investigación conocido como Computación Afectiva, mediante el cual se pretende desarrollar sistemas capaces de reconocer, modelizar, simular y comunicar estados emocionales. El fin perseguido es simular empatía en los robots, permitiéndoles interpretar el estado emocional de los humanos con los que interaccionan y adaptar su comportamiento a ellos, siendo coherentes en dicha interpretación.

Inspirados por la Computación Afectiva y la Neuroenconomía, consideramos que usar componentes afectivos como un elemento esencial en el proceso de toma de decisiones puede llevarnos a mejorar la interacción entre los agentes robóticos y los usuarios. Nuestro propósito es hacer a un agente más atractivo al interaccionar con él y, por lo

tanto, prolongar su uso por parte de los usuarios. La inclusión de estos modelos facilitaría su utilización en aplicaciones como mascotas de compañía dando apoyo emocional a personas de la tercera edad o a niños con autismo, o como apoyo educacional en un entorno de entretenimiento educativo.

Nuestra motivación para realizar esta tesis es el diseño de modelos afectivos de toma de decisiones y su implementación en robots domésticos de bajo coste. Dichos robots formarán sociedades donde interactúan entre ellos y con uno o más usuarios.

A lo largo de esta tesis describimos modelos de comportamiento basados en el Análisis de Riesgos Adversarios para un agente decisor autónomo. Dicho agente procesa información procedente de sus sensores enfrentándose a varios adversarios inteligentes. Los modelos desarrollados han sido implementados en un robot AiSoy1, una plataforma económica dotada de varios sensores y actuadores. Como veremos, su capacidad de procesamiento nos invita a desarrollar modelos simples pero robustos y rigurosos. La implementación en el robot se ha llevado a cabo utilizando el sistema operativo ROS y varios módulos propios desarrollados para interactuar con los sensores y actuadores sobre la plataforma.

La toma de decisiones competitiva se ve tradicionalmente desde la perspectiva de la teoría de juegos no-cooperativos, bajo la aceptación de la hipótesis de conocimiento común, según la cual, los agentes participantes conocen las probabilidades, utilidades y alternativas de cada uno de ellos, así como que toda esa información es, a su vez, conocimiento común. Los agentes persiguen un equilibrio de Nash, o alguna solución similar. Sin embargo, se ha demostrado que la aceptación de dicha hipótesis puede resultar poco apropiada en situaciones reales.

El Análisis de Riesgos Adversarios (ARA) es una disciplina reciente en el marco de la toma de decisiones competitivas que nos permite evitar suponer importantes hipótesis, como la mencionada del conocimiento común, usando modelos bayesianos sobre las creencias y preferencias de los oponentes. A través de ARA, apoyamos a uno de los participantes, en este caso un robot, resolviendo un problema de análisis de decisiones, utilizando procedimientos que emplean la estructura del problema del adversario para asignar probabilidades sobre las acciones de los oponentes. Las primeras aplicaciones de ARA han sido en áreas como el contraterrorismo y la ciberseguridad, entre otras.

Inicialmente, hemos utilizado el análisis de riesgos adversarios para diseñar un modelo de toma de decisiones que apoye a un agente autónomo que toma decisiones interactuando con un usuario. Tanto el agente como el usuario realizan acciones definidas en un conjunto finito, e interactúan en un entorno que varía dependiendo de las acciones del usuario. El agente dispone de varios modelos de predicción que le permiten estimar cómo variará el entorno y cuál será la próxima acción realizada por el usuario, dependiendo de si éste es considerado como reactivo o independiente de las acciones del agente. Por lo tanto, dispondrá de varios modelos de predicción, que se combinarán mediante técnicas de *medias de modelos*. Consideramos que nuestro agente se enfrenta a adversarios no estratégicos, por lo tanto, según la terminología ARA, nos encontraremos en un nivel-1 de la jerarquía mencionada, asumiendo que los oponentes se localizarán en el nivel-0.

A través de sus sensores inferirá, determinística o probabilísticamente según corresponda, cómo ha evolucionado el entorno y cómo se ha comportado el usuario. Mediante un modelo de preferencias evaluará el impacto que el entorno y la acción del usuario han provocado en sus cinco objetivos, que están ordenados jerárquicamente obedeciendo a sus necesidades vitales, i.e. el objetivo más importante, tendrá un peso mayor. Dicha evaluación se realiza con una función de utilidad aditiva multiatributo. Resolviendo un problema de optimización, el agente decidirá la próxima acción a realizar. Debido a las limitaciones computacionales a las que nos enfrentamos, el modelo de optimización estará limitado a planificar un único instante de tiempo hacia adelante.

Se han realizado dos simulaciones para estudiar la validez de estos modelos. En la primera, evaluamos satisfactoriamente los modelos de predicción desarrollados. En la segunda, el agente se enfrenta a dos tipos de usuario, uno malévolo y otro benévolo. Mediante estas simulaciones, demostramos que el diseño e implementación de modelos ARA es factible en un contexto robótico, permitiendo que éstos interactúen de forma realista con diferentes tipos de usuario, adaptando sus decisiones al comportamiento de cada uno de ellos.

Tras diseñar el comportamiento de un agente robótico, exploramos sus necesidades sociales y cómo tratar las situaciones en las que varios agentes compiten entre ellos y en sus interacciones con varios usuarios, o cooperan para satisfacerse a sí mismos y a los usuarios. Para ello, primero extendemos el modelo básico para permitir al agente interactuar con varios adversarios, agentes o usuarios. Mediante esta extensión, condicionamos los modelos de predicción al usuario al que estimamos que nuestro agente se está enfrentando, y ampliamos el modelo de preferencias para contemplar todas las posibles situaciones.

En aquellos escenarios donde los agentes sean competitivos, exploramos cómo nuestro agente resolvería dichas situaciones bajo el marco de la teoría de juegos, asumiendo ciertas hipótesis, y bajo ARA, que nos permite no asumirlas. Describimos dos de los múltiples escenarios que podemos encontrarnos. En el primero de ellos, consideramos el caso en el que tanto agentes como usuarios comparten el mismo entorno e interaccionan entre sí. En el segundo suponemos que cada agente apoya en la toma de decisiones a uno de los usuarios, formando así un equipo usuario-robot, que competirá contra otros equipos. Los agentes conocen lo que otros agentes hacen, pero no lo que el resto de usuarios hace.

En el caso de situaciones cooperativas, apoyamos a una sociedad de agentes, en lugar de a un único individuo. En estos escenarios, varios agentes colaboran entre ellos para encontrar la mejor solución que les satisfaga globalmente en su ayuda a los usuarios. Presentamos una nueva solución para juegos cooperativos en entornos finitos, consistente en maximizar la distancia al punto de desacuerdo, que será la solución competitiva. Además, demostramos que dicha solución cumple los axiomas de eficiencia de Pareto, simetría e independencia frente a alternativas irrelevantes, comparándola con las soluciones de Nash y Kalai-Smorodinsky.

Como estamos tratando con agentes autónomos, esperamos que sean ellos quienes decidan si quieren cooperar o competir. Para ello, desarrollamos un modelo paramtrico incluyendo parámetros que regulen la cooperatividad y competitividad de los agentes, los cuales dependerán de la experiencia previa del agente o de su estado. Cada uno de los agentes dispondrá de unos parámetros individuales que serán enviados a un ente computacional externo encargado de calcular el grado de cooperatividad de la sociedad y proponer una solución dado dicho parámetro. Dependiendo del grado de cooperatividad de la sociedad, la solución tenderá a maximizar o a minimizar la distancia a la solución competitiva.

Tanto las situaciones competitivas como las cooperativas, han sido validadas mediante simulaciones. En el caso de las competitivas, demostramos cómo un agente de nivel-1 en la jerarquía ARA se comporta enfrentándose primero a otro agente de nivel-1 y, después, a uno de nivel-0. Cuando todos los miembros de una sociedad de agentes realizan un análisis de nivel-1, alcanzan, tras un período razonable de tiempo, un equilibrio de Nash similar a los que podríamos encontrar en situaciones de juego ficticio. Por otro lado, podemos extraer de las simulaciones, que un agente decisor estratégico de nivel-1, al enfrentarse a agentes con modelos de predicción similares, obtendrá niveles mayores de utilidad que un agente no estratégico, sin embargo, valores más bajos de utilidad esperada.

En cuanto a los casos cooperativos, hemos simulado dos agentes interactuando entre sí y con un usuario. Ambos agentes estaban dispuestos a cooperar dependiendo de sus parámetros. Para poder realizar comparaciones en situaciones similares, cada agente calculaba su propia solución competitiva. Basándonos en los resultados obtenidos, observamos que la cooperatividad o competitividad de la sociedad de agentes, afecta a las acciones realizadas por cada uno de los agentes involucrados. Por otro lado, al comparar el comportamiento de un agente completamente competitivo, usando modelos ARA, con el de un agente más dispuesto a cooperar, podemos apreciar que el segundo agente obtiene valores mayores en la utilidad obtenida de las consecuencias, y en la utilidad esperada antes de tomar una decisión.

Consideramos que en un futuro cercano, los robots formarán parte de nuestras vidas diarias, por lo tanto necesitaremos que sean lo más interactivos posible, incluyendo mecanismos sociales y afectivos que proporcionen interacciones hombre-robot naturales y satisfactorias. Por dicho motivo, hemos diseñado un modelo que complemente a los de toma de decisión descritos previamente, otorgando a un agente autónomo la capacidad de decidir siendo influenciado por componentes afectivos al interactuar con humanos u otros agentes.

La mayoría de los modelos revisados en la literatura requieren una capacidad de cálculo elevada no disponible en plataformas domésticas. Por ello, hemos limitado nuestro modelo a dos componentes afectivos principales: emociones y estado de ánimo. Este último influye dinámicamente los pesos en el modelo de utilidad esperada multiobjetivo. Además, incorporamos la posibilidad de que el agente se comporte de forma impulsiva cuando alguna de las emociones alcance intensidades muy altas. En la literatura podemos encontrar otros componentes afectivos con gran relevancia a la hora de diseñar agentes sociales y emocionales, como la personalidad, la actitud o los estados motivacionales. Debido al incremento en complejidad y la poca relevancia que conllevaría el incluirnos en nuestro modelo, los hemos apartado por el momento.

Utilizamos cuatro emociones básicas: dos anteriores a la toma de decisiones (esperanza y miedo), y dos posteriores (alegría y tristeza). Para modelizar dichas emociones, consideramos variables como la expectativa, definida por la utilidad esperada asociada a la acción óptima; la incertidumbre respecto a la expectativa, definida por la varianza de la utilidad para la alternativa óptima; la deseabilidad de la consecuencia, definida por la utilidad obtenida en un determinado momento; y la sorpresa entre el suceso ocurrido y el esperado, definida por una función que mide la distancia entre distribuciones de probabilidad a priori y a posteriori.

El estado de ánimo se considera como un factor con impacto en los procesos cognitivos y de comportamiento de individuos. En los modelos descritos anteriormente, las preferencias del agente eran estáticas, en el sentido de que los pesos asociados a cada objetivo del agente eran constantes. A través de este modelo, influenciamos dichos pesos mediante el estado de ánimo, variando así el impacto que la evolución del entorno o el comportamiento del usuario tengan sobre los objetivos del robot. Además, en cada interacción con el usuario, nuestro agente expresará aquella emoción que tenga mayor intensidad, a través de su expresión facial y de un color, haciendo de este modo, la interacción entre el usuario y la máquina más realista y natural.

Hemos simulado situaciones en las que nuestro agente afectivo interactúa con usuarios con comportamientos diferentes, comparando su actuación con la de un agente sin componente afectivo como el descrito en el modelo básico. Podemos concluir que: la evolución de las emociones y del estado de ánimo difiere dependiendo del comportamiento del adversario al que el agente se enfrenta; y que el agente influido por componentes afectivos, obtiene mejores resultados, en términos de utilidad esperada y utilidades alcanzadas tras las consecuencias, que el agente básico, al interactuar con un agent con un comportamiento malévolo. De esto modo, muestra mayor versatilidad al interactuar con diferentes adversarios, adaptando su comportamiento según lo necesite.

Chapter 1

Machines that Perceive, Feel and Decide

1.1 Our interest in assistant machines

From primitive machines to autonomous robots

> "There is only one condition in which we can imagine managers not needing subordinates, and masters not needing slaves. This condition would be that each instrument could do its own work, at the word of command or by intelligent anticipation, like the statues of Daedalus or the tripods made by Hephaestus, of which Homer relates that "Of their own motion they entered the conclave of Gods on Olympus", as if a shuttle should weave of itself, and a plectrum should do its own harp playing."

This quote belongs to the great Aristotle (384 BC - 322 BC), who, in his *Politics* (first book, chapter II, On Slavery), speculated about the concept of autonomous machines and its potential applications, bringing someday human equality.

Humankind has indeed been interested in building machines that could help us and make our lives easier for a long time. In fact, the first presence of such machines is associated with ancient China, where, during the 6^{th} century BC, the first water clocks were designed to be used as a stop-watch for limiting the time on clients' visits to brothels, see Landels (1979).

One of the pioneers in the automata field was Al-Jazari (1136 - 1206), a Muslim engineer, inventor, artist and mathematician who lived during the Artuqid dynasty. He designed and built several automatic machines, such as the musical robot band, see Fig. 1.1(a). This robot consisted of a boat with four automatic musicians that floated on a lake to entertain guests at royal parties. Its mechanism was powered by water and was programmable to interpret melodies with different rhythms. In his widely quoted "Book of Knowledge of Ingenious Mechanical Devices" (1206) he collected many of his ideas, pioneering the field of mechanical engineering, see Figs. 1.1(b) and 1.1(c), in which two of his most famous inventions, the Elephant Clock and a hydropowered water-raising machine, are shown respectively.

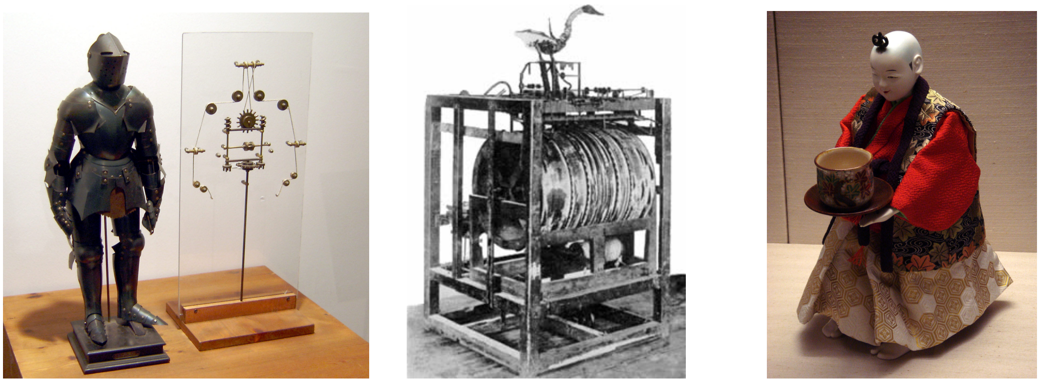
(a) (b) (c)

Figure 1.1: (a) The musical robot band. (b) The Elephant Clock. (c) The hydropowered water-raising machine.

Two hundred years later, one of the first designs of a humanoid robot was recorded: the mechanical knight invented by Leonardo da Vinci (1452 - 1519). As it was found in Da Vinci's notebooks, this mechanical knight was able to sit down and stand up, wave its arms and move its head and jaws, thus being one of the first defensive automata, see Fig. 1.2(a) for a recent implementation of Leonardo's ideas.

With less combative emphasis, Jacques de Vaucanson (1709 - 1782) built an automaton flute player and a tambourine player. In 1737, he built *Le Canard Digerateur*, see Fig. 1.2(b), which, powered by weights, was capable of imitating a real duck by flapping its wings, eating grain, digesting it and even defecating it.

Far from Europe, the Japanese craftsman Hisashige Tanaka, known as “Japan’s Edison”, built in 1796 a set of extremely complex mechanical toys which were used to serve tea, see Fig. 1.2(c), fire arrows drawn from a quiver or paint Japanese kanji characteres.



(a) (b) (c)

Figure 1.2: (a) Leonardo’s knight. (b) Le Canard Digerateur. (c) A japanese tea-servant toy.

Apart from designing human and animal-shaped automata, scientists were interested in building machines to help them with mathematical problems. Blaise Pascal invented the mechanical calculator in 1642, which could be used to add and subtract two numbers automatically. He was followed by Giovanni Poleni who built the second functional mechanical calculator in 1709, which could multiply two numbers automatically. In the beginning of the 20^{th} century, the idea of programmable machines became popular with analogue calculating machines. Leonardo Torres Quevedo (1852 - 1936), one of the most brilliant Spanish scientists, designed and built a whole series of analogue calculating machines, all mechanical, seeking for solutions to various mathematical equations.

Some years later, in 1936, Alan Turing (1912 - 1954), proved that machines would be capable of performing any conceivable mathematical computation if it were representable through an algorithm, see Turing (1937), thus creating the basis for what is now called *computer science*. In 1950, he proposed the Turing test opening with the words: “I propose to consider the question: can machines think?”. Such test evaluates a machine’s ability to exhibit intelligent behavior, so that a human engages in a natural language conversation with another hu-

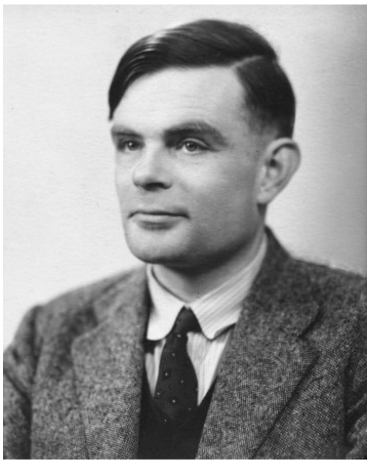


Figure 1.3: Alan Turing, the *father* of Computer Science.

man and the machine evaluated. The test will be passed if the human judge is unable to identify which of the conversations corresponds to the human and the machine, see Turing (1950). Until today, no machine has been able to pass it successfully. Following Turing test's steps, in 1990, an annual competition to find the most human-like chatbot (a computer program designed to simulate intelligent conversations) was released, known as the Loebner Prize.

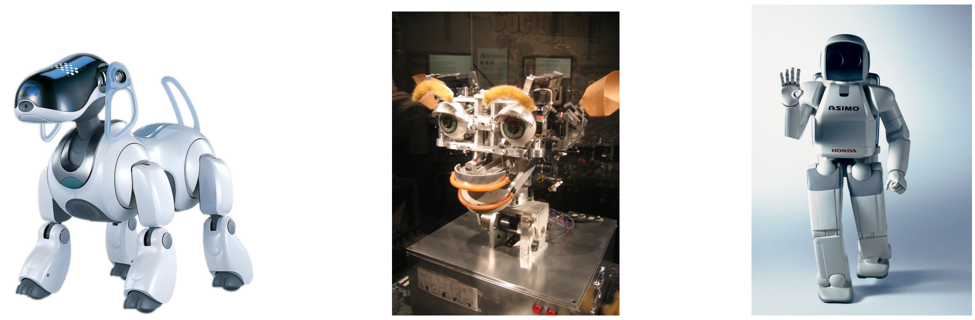
The industrial field has been where robotics has had the highest impact on assisting humans in their work. Industrial robots are programmable machines capable of moving parts or objects through a pre-specified sequence of motions. Such robots can have a great precision and repeat the same task for a long period of time. In 1961, the first industrial robot began to work at a General Motors assembly line. It was known as Unimate, and designed by George Devol.

The first electronic autonomous robots with complex behavior were created by William Grey Walter. He aimed at proving that the reason of how the brain worked was in how it was wired up. His first robots, named Elmer and Elsie, were built between 1948 and 1949 and were described as *tortoises* due to their shape and slow speed. They were capable of *phototaxis*, through which they could find their way to a recharging station when they ran low on battery power.

In 1989, due to innovative quick and cheap contruction methods, an hexapodal robot named Genghis was presented by the MIT. Genghis was famous for being made by Brooks & Flynn (1989) who published "Fast, Cheap, and Out of Control:

A Robot Invasion of The Solar System", in which they advocate for creating smaller and cheaper robots to decrease the difficulty of launching robots into space.

In order to make computers and automata more popular, in 1997, IBM announced a chess program called Deep Blue which was capable of defeating the World Chess Champion Garry Kasparov playing at the "Grandmaster" level. This super computer was capable of processing twice as many moves per second as it had during the first match (which Deep Blue had lost). The event was widely broadcasted live over the internet and received over 74 million hits.



(a) (b) (c)

Figure 1.4: (a) AIBO. (b) Kismet. (c) Honda's Asimo.

A year later, Sony introduced AIBO, see Fujita & Kitano (1998), becoming the first domestic commercial robot. It was released in Japan and sold out in 20 minutes. It was a robotic dog capable of interacting with humans. Using its LED screen, see Fig.1.4(a), it was capable of expressing emotions, depending on how the user interacts with it. The robot was allowed to learn from the user, the environment and other robots, in order to grow up from being a puppy to becoming an adult. Besides being used as a robotic pet, AIBO was widely used in research centers, because of its versatility. This robot had a huge impact in the robotic field and on the way that society started looking at robotics. Kismet was the first social robot ever built. It was designed by Cynthia Breazeal at MIT in 2000, see Fig. 1.4(b), and aimed at interacting and communicating with humans following social behaviors. The same year, Honda presented their humanoid project ASIMO, a robot capable of running, walking, communicating with humans, undertaking facial recognition, etc, see Fig. 1.4(c), demonstrating numerous potential applications of robotics. Its main objective was to assist people who lacked full mobility. This robot has made public appearances around the world. Indeed, in the USA, ASIMO is part of an attraction at Disneyland featuring a 15-minute show called “Say 'Hello' to Honda's ASIMO” since June 2005.

Robotics made its impact on the domestic market through a robotic vacuum cleaner called Roomba in 2004, designed by iRobot, led by Rodney Brooks. Others robots that have been introduced into the domestic market are automatic pool cleaners or toys like Furby, but their intelligent behavior is frequently questioned.

Following ASIMO steps, but at a smaller scale, NAO was presented in 2004. It has became one of the most used research platforms due to the amount of actuators and sensors it has. It is the current platform used in the RoboCup competition, see Section 1.2.2. It has been used with lots of purposes. For example, in the summer of 2010, Nao made a synchronized dance routine at the Shanghai Expo in China, whereas in December 2010, a Nao robot did a stand-up comedy routine.

After Deep Blue's success, IBM decided to create a machine capable of answering questions posed in natural language. It was called *Watson*. In 2011, Watson competed in a TV show called *'Jeopardy!'*, against former winners receiving the first prize. Such show consisted on activating a buzzer, in order to respond a question given a clue, before other players active their corresponding buzzers.

Our platform: AiSoy1

AiSoy1 is the first robot manufactured by AiSoyRobotics (2010), see Fig. 1.5. It belongs to the robotic pets field, but it has numerous potential applications in domotics, eduitainment, therapeutic environments and research. It is a robot capable of making decisions depending on the emotion is simulating. These emotions will be affected by the impact of recently occurred events, as being attacked by the user, or being warm and comfortable at the room where it is. The originally implemented emotional model was based on El-Nasr *et al.* (2000)'s FLAME model, but the robot ended up too emotionally unstable.

AiSoy1 is a very interesting research platform, due to the variety of sensors and actuators included in it. It works with a Raspberry Pi Model B, see Raspberry Pi (2011), using a Linux based operating system known as AiROS.

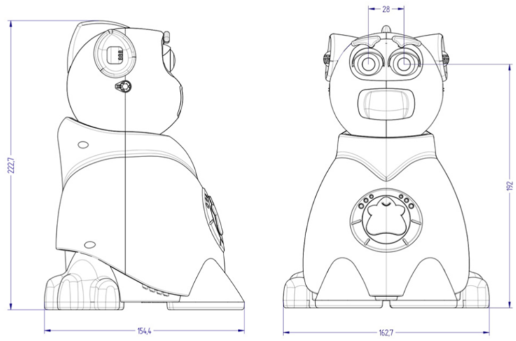


Figure 1.5: AiSoy1.

This robot has several sensors including a camera to detect objects or persons within a scene; a microphone used to recognize when the user talks and understand what she says, through an ASR (Automatic Speech Recognition) component; several touch sensors to interpret when it has been stroked or attacked; an inclination sensor so as to know, whether or not, the robot is in vertical position; a light sensor and a temperature sensor. As actuators, it includes several servos that allow it to move some parts of its body, but it mostly uses a text-to-speech system (TTS) combined with a led matrix to simulate a mouth when talking. Using a RGB led in the middle of its body, it is capable of showing different colors that symbolize the predominant emotion at that moment, see Fig. 1.5. It may use a radio module to communicate with other robots. The information provided by these sensors is processed by the robot to detect the user and infer the user's actions and the environmental state, as described in Section 2.4.1.

The ASR component used within the AiSoy1 is PocketSphinx, a version of CMU Sphinx that can be used in embedded systems developed at Carnegie Mellon University, see Carnegie Melon (2013). This component is constantly listening to what is surrounding the agent. Whenever it identifies a word with a probability above a certain threshold, it is sent to the interpreter module which is based on ChatScript, see Wilcox (2013), which, also probabilistically, determines the set of words that the identified word belongs to.

1.2 Intelligent agents and multi-agent systems

In this Section, we describe the main concepts related with intelligent agents and multi-agent systems, focusing on social and emotional agents.

1.2.1 Intelligent agents

An intelligent agent is an autonomous entity which perceives, through sensors, and acts upon an environment, using actuators, directing its activities towards achieving goals. Intelligent agents may also have the capability of learning or using knowledge to achieve their goals. They may be very simple or very complex: a thermostat is an intelligent agent, as well as a human being is, see Russell (2003).

Our research is focused on computational intelligent agents, and specifically, on those which make decisions and interact with humans, also known as social agents. Fong *et al.* (2003) describe social agents as those that may exhibit some "human social" characteristics like: express/perceive emotional states, communicate with high-level dialogue, learn/recognize behavioral models of other agents, exhibit distinctive personality and character, among others.

A recent field called **Affective Computing**, see Picard (1997), aims at combining emotions and computers so as to develop systems which should be able to recognize, model, simulate and communicate emotions. The main motivation for researching in this field is to simulate empathy within the machines, in the sense of allowing them to interpret the emotional state of humans and adapt its behavior to them, being coherent with that interpretation. These affective computers should not only provide better performance in assisting humans, improving human-computer interaction (HCI), but also enhance computers' abilities in making decisions.

Some of the better known social and emotional robotic agents are:

Aibo is an entertainment robotic dog, see Fig. 1.4(a), developed and produced by Sony, see Fujita & Kitano (1998). It is mobile and autonomous, and has a programmable behavior. It is capable of *phototaxis*. Its main goal is to play and interact with humans, but it has had a remarkable importance as a research platform participating in experiments. It has been used extensively in studies with the elderly in order to try to assess the effects on the quality of life and symptoms of stress, see Broekens *et al.* (2009).

Kismet was the first design of a social robot, see Breazeal & Scassellati (1999, 2000). It was the first robot to face the challenge of interacting socially with a human. Quoting its authors:

> "One does not use Kismet to perform a task. Instead, Kismet is designed to be a robotic creature that can interact physically, affectively, and socially with humans in order to ultimately learn from them."

Kismet was modeled after an infant and is capable of proto-social responses, providing an untrained user with natural and intuitive means of communication.

PaPeRo was developed by NEC. It was designed aiming at a commercial product, see Fujita (2002), to help people interact with electronic devices around the house (e.g., TV, computer, answering service, etc.), see Fig. 1.6(a).

Pearl was one of the first nursing-assistant robots, see Pineau *et al.* (2003). This robot interacts with its surroundings through speech, visual displays, facial expressions and physical motion. Its design is aimed at assisting elder people. Due to the forgetfulness of that people, the need for a robot that can offer different reminders arises, so as the need to help elderly people walking, either to attend appointments, see Fig. 1.6(c).

iCat has been developed and produced by Philips Electronics, see van Breemen (2004). It has been designed aiming at being an interactive and believable companion robot in home environments. It has been used also as a research platform for human-robot interaction. It is able to express emotions throughout its cat-face servos, see Fig. 1.6(b).

Paro is a seal robot developed by Wada & Shibata (2007). It is designed using a baby seal as a model for its appearance, see Fig. 1.6(e), and is covered with real white fur. Its aim is to study the effects of Animal Assistive Therapy with companion robots, and is targeted at the elderly. It has programmable behavior as well as a set of sensors, but it is not mobile.

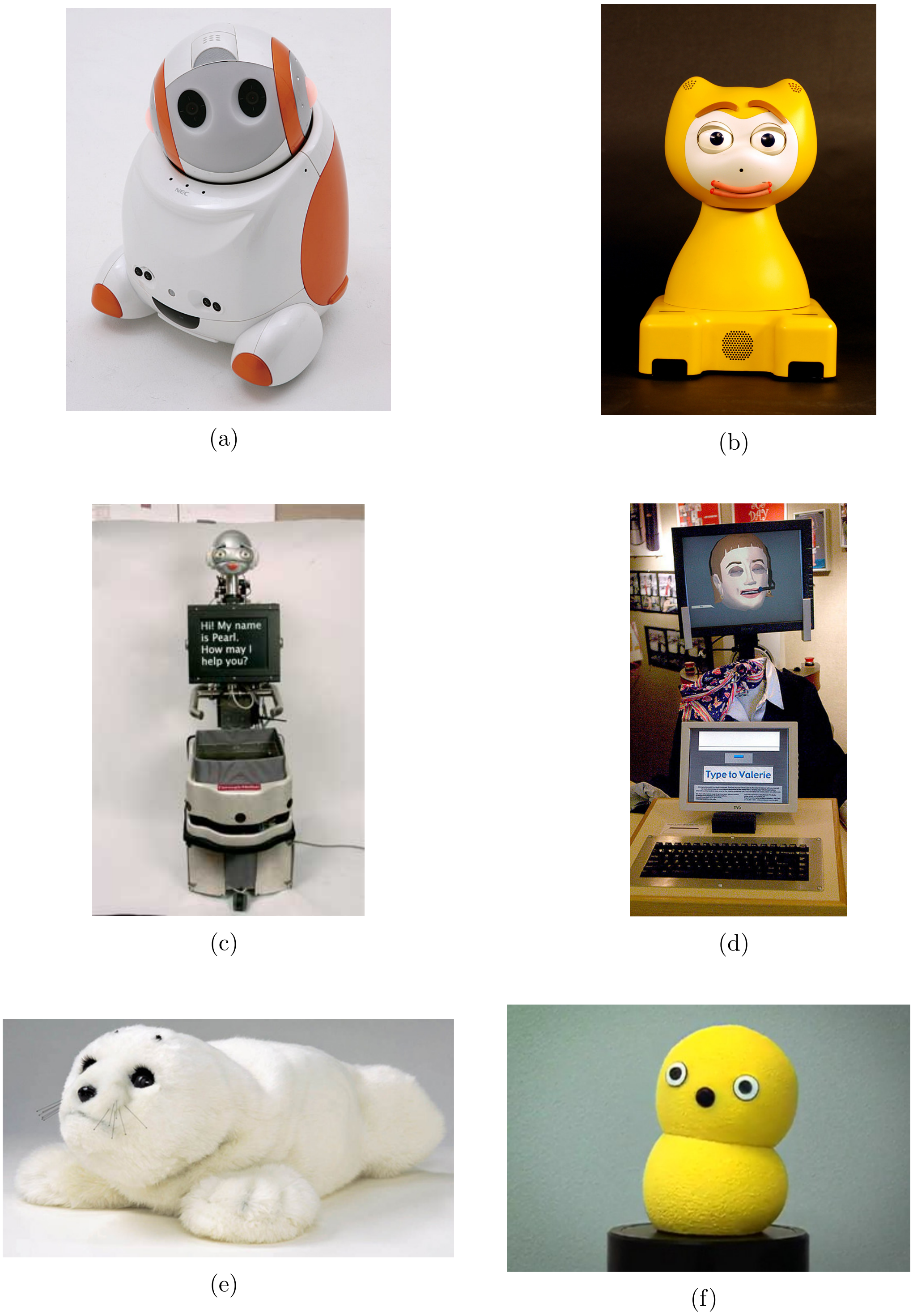
Keepon is a small creature-like robot, see Fig. 1.6(f), designed for simple, natural, non-verbal interaction with children, to help experts study, test, and elaborate psychological models of the development of social intelligence, see Kozima *et al.* (2009). The simple design of Keepon's appearance and behavior is meant to intuitively and comfortably convey the robots expressions of attention and emotion.

Roboceptionist was designed by Kirby *et al.* (2010), and it consists of an affective model for social robots which was implemented on a virtual robot face placed on a monitor, expressing its emotional states through animated facial expressions and a priori composed narrative, see Fig. 1.6(d). This model relies heavily on the designers who hard-wired the robot's background which is used to define its emotional state.

1.2.2 Multi-agent systems

Whenever we found several interacting intelligent agents, we may called that a Multi-Agent System (MAS). A MAS does not necessarily have to be compounded of intelligent agents. Indeed, a group of non-intelligent agents may form a MAS. Insect societies like ants, termites or bees, are multi-agent systems in which a collective intelligence, or *swarm intelligence*, emerges out of interactions among individual insects, see Bonabeau *et al.* (1997) and Beni & Wang (1989). The main factor of these multi-agent systems is that relatively simple individual rules can produce a large set of complex swarm behaviors.

In the context of this work, we are interested in autonomous and intelligent agents which work together in a MAS, building a society of self-interested intelligent agents. Agents in a multi-agent system cannot be assumed to share a common goal, as they will often be designed by different individuals in order to represent their interests. One agent's interests may therefore conflict with those of others, just as in human societies. Despite the potential conflict of interests,



(a) (b) (c) (d) (e) (f)

Figure 1.6: (a) PaPeRo, (b) iCat, (c) Pearl, (d) Roboceptionist, (e) Paro, and (f) Keepon.

the agents in a multiagent system will ultimately need to cooperate in order to achieve their goals, see Wooldridge (2008).

Traditional multi-agent systems research was concerned with problems of designing societies of autonomous agents, such as why and how agents cooperate, see Wooldridge & Jennings (1997); how agents can recognize and resolve conflicts, see Adler *et al.* (1989), Galliers (1988) and Galliers (1990); how agents can negotiate in situations where they are apparently hostile, see Ephrati & Rosenschein (1993) and Rosenschein (1994), and so on.

More recent MAS reseach is focused on areas like those mentioned next. MAS are widely used in agreement problems, where several parties receive a reward for reaching an agreement on some issue. They are generally solved through negotiation, see Endriss (2006) for a game theoretic perspective. Agreement problems are neither considered fully cooperative (due to agents having individual preferences), nor fully adversarial (because agents need to reach the agreement to receive the reward). They are found in the *semi-cooperative* field, in which agents are self-interested since they focus on their own goals, but they are also cooperative, as they are willing to collaborate with others to find solutions that are beneficial to all participants, including itself. As an interesting example on this field see Zhang & Lesser (2007), where agents do not want to voluntarily sacrifice their own interest for others.

The collaborative learning field aims at learning how to imitate the behavior of another agent by watching a demonstration of a particular task. Demonstration-based learning has been successfully applied to a variety of single agent learning problems, see Breazeal *et al.* (2004b) and Nicolescu & Mataric (2003). But this framework has been also applied to a collaborative multi-agent domain. Some examples are: Clouse (1996) which uses reinforcement learning in agents to request advice from other similar agents in the environment; Price & Boutilier (2003) present a multi-agent system in which novice agents learn by observing other agents in the environment; in Riley (2005), an external coach agent provides advice to a team of agents in order to improve their performance at a task.

Regarding team formation in heterogeneous teams, there are agents and robots with different capabilities. The term *synergy* is commonly used to describe how well team members work together. In Liemhetcharat & Veloso (2012) a model that captures synergy from the interactions among large groups of agents is introduced. The problem has been abstracted as finding the best subset of agents to complete a task. That approach models and learns from the interactions between large groups of agents to form an effective team.

There are some scenarios that are used as test-beds for multi-agent systems. RoboCup, see Asada *et al.* (1998), is the most extended test-bed for multi-robot cooperation and competition, but it requires domain-dependent models. It consists of a football match between two teams formed of autonomous robots. There exist soccer leagues, in which both cooperative and competitive behavior are performed. As an example see Phillips & Veloso (2009), who describe an scenario where members of the attacker team should cooperate to pass the ball and score a goal, while they should avoid members of the other team and any collision with them. The members of the defender team should predict and chase the ball to score.

A more abstract and simpler but versatile game is the Colored Trails (CT) Framework, see Grosz *et al.* (2004), in which players negotiate and exchange resources to enable them to achieve their individual or group goals. CT is played by two or more players on a board of colored squares, where each player is given a starting position, a goal position and a set of chips in colors. Players may move towards their goals by advancing to an adjacent board square, but only if the player has a chip of the same color as the square. Players may negotiate with their peers to exchange chips. It provides an analogue to the ways in which goals, tasks and resources interact in real-world settings, abstracting away the complexities of real-world domains. CT supports comparisons of the performance of different computational strategies for interacting in groups comprising people and computer agents as well as solely computer agents. In Gal *et al.* (2010), the CT framework is used to study several decision-making models based on social factors which influence people's decision-making, whereas in Gal *et al.* (2011), it is used to study how an agent adapts its behavior to negotiate with people from different cultures. A model of agents with the cognitive ability to make use of the theory of mind, to reason explicitly about the beliefs, desires, and goals of others, using the CT framework is detailed in Weerd *et al.* (2014).

1.2.3 Communication among robotic agents

To accomplish collaborative tasks, robotic agents need communication to coordinate. Team members may improve their understanding and the understanding of their teammates about the environment that surrounds them, by communicating their local information.

Agents can neither force other agents to perform some action, nor modify the internal states of other agents. What they can do is perform *communicative actions* in an attempt to influence other agents. Speech act theory treats communication as action, see Austin (1962). It lies on the assumption that speech actions are performed by agents just like other actions. After some extensions of Austin's work, see Searle (1969) and Cohen & Perrault (1979), it was observed that a theory of speech acts should be rooted in a more general theory of rational action. Cohen & Levesque (1988) develop a theory in which speech acts were modeled as actions performed by rational agents on behalf of their intentions.

Communication in multiagent systems is usually governed by **Agent Communication Languages**, see Kone *et al.* (2000) and Huget (2003), but in particular cases within heterogeneous multi-agent systems, the need for more flexible models arises, see Fischer *et al.* (2006) as an example.

The Knowledge Query and Manipulation Language (KQML) was one of the first Agent Communication Languages. It defines a format for messages, see Finin *et al.* (1994). Following the KQML protocol, each message should have a *performative* and a number of *parameters*. A performative is defined, in the speech act theory, as a sentence not only used to passively describe a given reality, but to change the reality described. To utter one of these sentences implies performing a certain kind of action, see Austin (1962). The performative will define the class of the message, and how the sender and the receiver shall deal with it. The parameters specify who is the sender, the receiver, what the sender expects to receive as a reply, etc. KQML has had a significant impact on the multi-agent systems community, and has been widely implemented and distributed, but was also criticized by the community. These criticisms led the Foundation for Intelligent Physical Agents (FIPA) to develop the FIPA ACL, see FIPA (1999). The main difference between the KQML and the FIPA ACL is the collection of performatives they provide. The FIPA ACL includes semantics allowing to represent beliefs and desires. Several platforms have been developed that support the FIPA ACL. The most-known and widely used is the Java Agent Development Environment (JADE), see Bellifemine *et al.* (2007).

Communication among agents is not cost-free. Multi-agent teams must trade off the benefit that can be achieved through communication with the cost of communicating, see Jim & Giles (2001). Once agents are allowed to communicate, they have to learn *what* and *when* to communicate with their teammates, see Roth *et al.* (2003) and Roth *et al.* (2005) respectively.

1.3 Games and decisions

In this Section, we briefly introduce some concepts related with the decision-making process upon which we should build on. We shall start with how to solve a decision-making problem by one agent. Then, through game theory, we shall see how an agent make decisions that may influence another's agent welfare, and some of the most debated problems that game theory implies. Next, we shall describe the recent framework of Adversarial Risk Analysis, which looks for supporting a decision-maker facing other decision-makers, as game theory does, but weakening some of its strong assumptions.

1.3.1 Decision theory

Decision theory is a discipline widely applied in fields like economics, psychology, mathematics or statistics. It is concerned with modeling values, uncertainties and other relevant aspects in a given decision, and obtaining the corresponding optimal decision. We may differentiate three approaches within decision theory. On one hand, the normative approach, concerned with identifying the best decision to make, assuming that we are supporting an ideal decision-maker who has full information, and is able to compute with perfect accuracy, being fully rational. On the other hand, we may find the descriptive approach, which assumes that actually there are no ideal cases, attempting to describe what people do. The third one is the prescriptive approach, in which advices are given to real people.

The field of decision analysis belongs to the prescriptive approach of decision theory. It aims at identifying, representing and assessing important aspects of the decision-making process, in order to prescribe a recommended course of action by applying the Expected-Utility Maximization Principle, see von Neumann & Morgenstern (1944).

1.3.2 Game theory: non-cooperative games

Decision analysis helps us in supporting a single agent facing a decision to make. There are some cases in which the agent interacts with one or more adversaries. In such areas, we may appeal to game theory, see Díaz *et al.* (2010). There are some basic assumptions that underlie game theory, see Myerson (1991), such as *rationality*, where players have some preferences over their payoffs, so that they are utility maximizers; or as the *common knowledge* assumption, which means that each player knows about his opponents' strategies, beliefs and payoff preferences, and each of them also knows that the other players know that he knows, and so on, *ad infinitum*.

Non-cooperative games are those in which players make decisions independently, and no compromise is allowed among the players. We can express the interactions among N adversaries, or players, as a game $(G = (N, (S_i)_{i \in N}, (u_i)_{i \in N}))$ defined by strategies $(s_i \in S_i)$ and payoff functions $(u_i \in U_i$, where $u_i : S \to R)$ for each player $(i \in N)$. There are two ways of representing these games: in strategic or in extended form.

Players' strategies S_i will determine the action that player i will take at any stage of the game, before the game starts. Players cannot communicate, so that we may suppose that they choose their strategies independently and simultaneously. Thus, we can view players' strategies as independent random variables. A *randomized strategy* for player i (τ_i) would be any probability distribution over the set of S_i, thus $\tau_i(s_i) \in \Delta(S_i)$. Strategies could be *pure or mixed*, due to player i could assess any probability $(\tau_i \in \Delta(S_i))$, so that in such cases in which $\tau_i(s_i) = 1$ or $\tau_i(s_i) = 0$, s_i would be a pure strategy.

Once we have seen how to represent games, we shall explain how to solve them using the **Nash Equilibrium** concept, see Nash (1951), which suggests that no player could increase his expected payoff by unilaterally deviating from the equilibrium point. That is, in any strategic-form game G, with N players, a randomized-strategy profile σ is a Nash equilibrium of G iff,

$$u_i(\sigma) \geq u_i(\sigma_{-i}, \tau_i),$$

for each strategy $\tau_i \in \Delta(S_i)$, and any player $i \in N$. Therefore, each player i would like to choose among those strategies that maximize his expected payoff respectively to those of the other players σ_{-i}.

$$\text{If } \sigma_i(s_i) > 0, \text{ then } s_i \in \arg\max_{d_i \in S_i} \sum_{s_{-i} \in S_{-i}} \left(\prod_{j \in N_{-i}} \sigma_j(s_j) \right) u_i(s_{-i}, d_i).$$

If the probability distribution of a strategy s_i is greater than zero, it implies that the strategy s_i belongs to the set of strategies that maximize the utility of player i for any strategy played independently by the rest of players (s_{-i}) in the game. Nash (1951) established that every strategic-form game has at least one equilibrium through the **General Existence Theorem**. The importance of randomized strategies lies in this theorem, because many games have no equilibria in pure strategies.

Despite the significant importance of the Nash Equilibrium concept, it could lead us to equilibria that are *Pareto inefficient*, or we can find games with *multiple equilibria* points. But we may conclude that the Nash Equilibrium concept would be a necessary condition for a theory to be a good prediction of the behavior of intelligent rational players, see Myerson (1991).

1.3.3 Bayesian games and other approaches

There are games in which some players have *private information* that is not common knowledge, see Díaz *et al.* (2010). That private information could be his beliefs about other's strategies, or about other's beliefs, etc. In realistic situations, players do not know each other payoffs so that they have to find a subjective probability distribution that accurately fits his beliefs about that.

Bayesian Games, see Harsanyi (1967), provides a methodology to gather all of those subjective probability distributions. It categorizes players through types. Depending on the amount of uncertainties in the game, the number of types will vary. The private information of each player fits with a type, and has some beliefs about the types the other players are. Those beliefs are computed using Bayes Theorem with a prior probability distribution over the set of types, that must be common knowledge among players.

To define a Bayesian game B, we should specify a set of players N and, for each player i in N, a set of possible actions C_i, a set of possible types T_i, a probability function p_i and a utility function u_i, thus $B = (N, (C_i)_{i\in N}, (T_i)_{i\in N}, (p_i)_{i\in N}, (u_i)_{i\in N})$. The probability function p_i is a function from T_i into $\Delta(T_{-i})$, so that for any possible type in T_i, it specifies a probability distribution $p_i(\cdot \mid t_i)$, representing what player i would believe about the other players' types if his own type were t_i.

As in classical games, rational players seek to maximize their expected payoff given their beliefs about the other players. Therefore, a **Bayes-Nash Equilibrium** would be defined as a strategy profile and beliefs determined for each player about the types of the other players, that maximizes the expected payoff for each player given their beliefs about the other players' types and given the strategies chosen by the other players, see Harsanyi (1967). When there are multiple Bayes-Nash equilibria it is the analyst who must decide which one to use.

Decision-Analytic Approach Through this approach, our task is to advise a particular player i to what strategy he should use in a given game, see Kadane & Larkey (1982) and Raiffa (1982), while in game theory is to analyze and solve the decision problems of all players together. To solve this approach, we should find player i's optimal decision, that would maximize his expected payoff with respect to his subjective probability distribution over the set of strategies of the other players. Despite this approach success weakening the common knowledge assumption, Kadane & Larkey (1982) did not specify how to assess a subjective probability distribution over the actions of other players, so this approach is incomplete, see Harsanyi (1982).

1.3.4 Game theory: cooperative games

In a cooperative game, players look for combining efforts to maximize their own benefits based on a collective utility function. There are several examples of cooperative games, including bargaining, arbitration or coalition games. We are mainly focused on bargaining games for the scope of our work.

We should remind that under the scope of game theory, each player is an intelligent and rational decision-maker, whose behavior is derived from the goal of maximizing his own expected utility payoff. Nash (1951) proposed that cooperation among players could be understood using the concept of cooperative Nash Equilibrium, assuming that cooperative actions are the result of a bargaining process where each player should be expected to behave according to some strategy that satisfies his personal utility-maximization criterion. In other words, in any strategic-form game G with multiple equilibria, some bargaining process could be used to allow players to communicate, or use arbitration to determine the focal equilibrium of the game. This focal equilibrium comes from the **focal-point effect**, see Schelling (1960), which describes that anything that tends to focus players' attention on a particular equilibrium, being that common knowledge, tends to make that point the equilibrium that the players will expect.

A formal definition of a bargaining cooperative game could be the following: N agents have access to any of the alternatives within some set, called *the feasible set* (F). Their preferences over these alternatives differ. If they agree on a particular alternative, that would be the solution of the conflict. Otherwise, they end up at a prespecified alternative in the feasible set, called the disagreement point (d). F tends to be a closed convex set of $\mathbf{R}^N$ in traditional cooperative game theory but it is not a requirement, there are also non-convex feasible sets, see Thomson (2009), within which there are finite feasible sets, see Wu (2007), which are of our interest for this thesis, see Chapter 3. The disagreement point d is a vector in F. Thus, a bargaining problem is defined by the tuple $(F,\ d)$, and would be solved by a solution concept $\phi(F,\ d)$, see Díaz *et al.* (2010) for a good introduction to cooperative game theory.

To find solution concepts ($\phi(F,\ d)$) for this kind of games, there exists several options including Nash's, Kalai-Smorodinsky's, Yu's, Egalitarian, Dictatorial and

Raiffa's. We shall describe the first three solution concepts. The axiomatic characterization of each solution is not detailed here as, in Chapter 3, we provide a discrete approach to them.

The Nash solution: Nash (1950) proposed a solution concept which suggests choosing the agreement that maximizes the product of gains, see $\phi_N(F, d)$ in Fig. 1.7:

$$\phi_N(F, d) \text{ is the maximizer of } \prod(x_i - d_i) \text{ for each } x \in F.$$

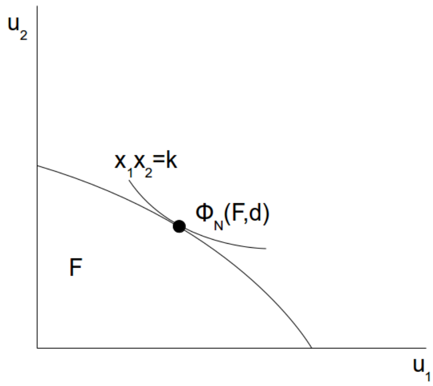


Figure 1.7: Nash Bargaining Solution.

The Kalai-Smorodinsky solution: This alternative, introduced by Kalai & Smorodinsky (1975), finds the solution in the straight line, that goes from the disagreement point (d) to the utopia point ($u^*(F, d)$), which is the point where both players would maximize their own utilities simultaneously. Whenever that line crosses the Pareto frontier, see $\phi_K(F, d)$ in Fig. 1.8, the solution would be found. $\phi_K(F, d)$ requires convexity of the feasible set F.

This solution concept is the maximal point ($m_i(F, d)$ for each player i) of F on the segment which connects the disagreement point to the utopia point ($u^*(F, d)$), defined by $u_i^*(F, d) = max\{x_i \mid x \in F, x \geq d\}$ for all i. In other words, in a two-person bargaining game

$$\phi_K(F, d) \text{ is the maximizer of } \frac{x_2 - d_2}{x_1 - d_1} = \frac{m_2(F, d) - d_2}{m_1(F, d) - d_1}.$$

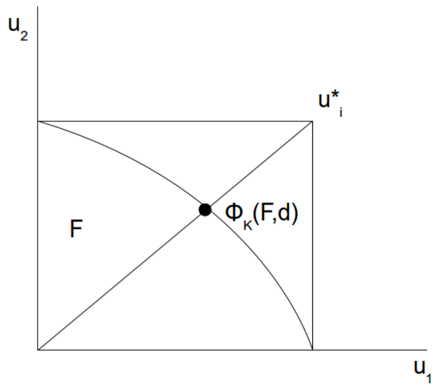


Figure 1.8: Kalai-Smorodinsky Solution.

Yu's solution: This is not one of the better known alternatives but we describe it for its later relevance in Chapter 3. It was introduced by Yu (1973), and it is aimed at looking for the best decision in which players minimize their regret in terms of their utility functions, see $\phi_Y(F, d)$ in Fig. 1.9. $\phi_Y(F, d)$ requires convexity of the feasible set F. $D_p(u(x))$ is called the *group regret* for decision x, and it is expressed through

$$D_p(u(x)) = \left[\sum_{i=1}^{n} (u_i^* - u_i(x))^p \right]^{\frac{1}{p}},$$

with $1 \leq p \leq \infty$, a parameter that serves as "balancing factor". For example, when $p = 1$, $D_1(u(x))$, we get the compromise solution which minimizes the sum of all individual regrets $|u_i^* - u_i(x)|$ known as "group utility". When $p = \infty$,

$$D_\infty(u(x)) = \max_i \{u_i^* - u_i(x) | i = 1, 2, \ldots, n\},$$

we get the compromise solution, which minimizes the maximal regret an individual may have. In Freimer & Yu (1976), it is said that only in two-players game, when the balancing factor p is small, the "group utility" is emphasized and when p increases, it is the "individual utility" which receives more weight.

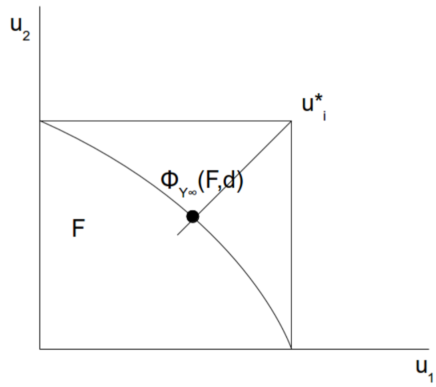


Figure 1.9: Yu Solution.

1.3.5 Adversarial risk analysis

Competitive decision-making is traditionally viewed from the perspective of non-cooperative game theory, see e.g. Gibbons (1992), Myerson (1991) or Menache & Ozdaglar (2011). Under a strong assumption of common knowledge participating agents know each other probabilities, utilities, and choice sets, and all know that these are common knowledge, as explained above. Then, agents aim at computing Nash equilibria or related refinements. However, such common knowledge assumption is implausible in many applications. Raiffa *et al.* (2007), Rothkopf (2007), and Lippman & McCardle (2012) discuss this in detail, pointing out to other defects like there being several equilibria with no unambiguous criteria to further discern among them.

Adversarial Risk Analysis (ARA) is a recently introduced framework characterized by the fact that there are two or more intelligent opponents who make decisions for which the outcome is uncertain and independent, see Ríos Insua *et al.* (2009). It avoids the common knowledge assumption through an explicit Bayesian model of the capabilities, probabilities and utilities used by the opponents.

ARA aims at supporting one of the participants viewing the problem as a decision analytic one, using principled procedures that employ the adversarial structure to assess the probabilities on the opponents' actions, and finally, applying decision analysis to maximize expected utility. This approach has a Bayesian game-theoretic flavour, see Kadane & Larkey (1982), and Raiffa *et al.* (2007).

ARA was motivated by recent applications in counterterrorism, cybersecurity and competitive corporate decision-making. For various concepts, methods and applications see Ríos Insua *et al.* (2009), Wang & Banks (2011), Banks *et al.* (2011), or Ríos Insua *et al.* (2012).

Under an ARA strategy, a player acts in order to maximize its expected utility under subjective beliefs about the probabilities of its opponents' actions and utility functions. This subjective belief is developed using a hierarchy of nested models for the decision processes, following a Bayesian version of level-k thinking, see Stahl & Wilson (1995). This k-level hierarchy is indexed by how deep the player thinks its opponents' decision-making process are. If the opponents do not follow any strategy we would be dealing with a zero-level model; if the opponents look for modeling the player's thinking, then it is a first-order analysis model; if the opponents model the player's model of the opponents' decision-making, then it is a second-order analysis model, and so forth.

The main obstacle in other approaches that follow the same Bayesian strategy, as pointed out in Kadane & Larkey (1982) and Harsanyi (1982), is to find the correct mechanism that allows a decision-maker to develop subjective probability distributions which adequately represent an opponent's behavior. ARA tries to resolve that through a "mirroring" procedure, in which the decision-maker does the analysis that he expects his opponent to make, considering so far the fact that the opponent may be simultaneously performing an analysis of the decision-maker's process.

Suppose a decision-maker called *Apollo*, and his adversary *Daphne*. Both of them have a finite set of actions (A and D), utility functions ($u_A(\cdot)$ and $u_D(\cdot)$), a collection of probabilities about outcomes (P_A and P_D) and expected utilities (ψ_A and ψ_D). The utility that Apollo expects from performing action $a \in A$ when Daphne choose action $d \in D$ is:

$$\psi_A(a, d) = \int u_A(c)\pi_A(c|a, d)dc,$$

where $\pi_A(c|a, d) \in P_A$ models Apollo's beliefs about the consequences ($c \in C$) for the pair of actions (a, d). The expected utility for Daphne ($\psi_D(a, d)$) would be analogous.

In a non-cooperative game-theoretic scenario, $\Psi_A(a,d)$ and $\Psi_D(a,d)$ would be used to find the Nash Equilibrium (a^* and d^*) satisfying

$$\psi_A(a^*,d^*) \geq \psi_A(a,d^*)\ \forall a \in A,$$

$$\psi_D(a^*,d^*) \geq \psi_A(a^*,d)\ \forall d \in D.$$

Let us suppose now that we would like to support one of the players (Daphne) against the other (Apollo) in a simultaneous decision game.

The solution of that game would be:

$$d^* = \arg\max_{d\in D} \sum_{a\in A} \left[\sum_{s\in S} u_D(d,s) p_D(S=s|a,d)\right] \pi_D(A=a),$$

where S is an uncertain outcome that represents Apollo's success, and $p_D(S|a,d)$ would be Daphne's beliefs about Apollo's success given Apollo's and Daphne's actions. As mentioned above, the main obstacle is the assessment of $\pi_D(A=a)$. To do so, Daphne would solve Apollo's decision analysis:

$$a^* = \arg\max_{a\in A} \sum_{d\in D} \left[\sum_{s\in S} u_A(a,s) p_A(S=s|a,d)\right] \pi_A(D=d).$$

Using a Bayesian strategy, we should put a distribution over Apollo's $(u_A, p_A, \pi_A) \sim (U_A,\ P_A,\ \Pi_A)$ which would be estimated by Monte Carlo simulation, see Ríos Insua *et al.* (2009). Considering Apollo as a rational player, Daphne would model Apollo's decision problem as

$$A \mid D \sim \arg\max_{a\in A} \sum_{d\in D} \left[\sum_{s\in S} U_A(a,s) P_A(S=s|a,d)\right] \Pi_A(D=d).$$

The elicitation of $\Pi_A(D=d)$ would require further analysis at the next level of hierarchy thinking. Assuming that Daphne considers that Apollo is thinking of Daphne as a strategic thinker, she would solve

$$D \mid A^1 \sim \arg\max_{d\in D} \sum_{a\in A} \left[\sum_{s\in S} U_D(d,s) P_D(S=s|a,d)\right] \Pi_D(A^1=a).$$

It could lead to an infinite regress thinking-about-what-the-other-is-thinking-about ... Therefore, we should stop when we have no more information about utilities and probabilities at some level of the recursive analysis.

1.3.6 Social decision-making models

Through this Section, some decision-making models based on social preferences are introduced. Among all the studied social factors included in social decision-making models, we describe those which are considered the most relevant ones, see Fehr & Schmidt (2006):

Altruism: An altruistic agent is willing to sacrifice its own payoff in order to improve the well-being of others, see Andreoni & Miller (2002) for a model based on this social factor.

Envy: It is the opposite case of altruism. An agent of this kind is always willing to decrease the payoff of a reference agent at a personal cost to itself. See Bolton (1991) and Kirchsteiger (1994) for some reference models based on it.

Inequity Aversion: An agent is inequity averse if, in addition to its self-interest payoff, its utility increases if the allocation of payoffs becomes more equitable. The controversy arises with the concept of *equitable*, which has been defined in various ways. Fehr & Schmidt (1999) and Bolton & Ockenfels (2000) provide robust models based on this concept.

Reciprocity: A reciprocal agent responds to kindness in a kind manner, and to actions considered to be hostile in a hostile manner. Rabin (1993), Dufwenberg & Kirchsteiger (2004) and Falk & Fischbacher (2006) are relevant models based on this concept.

We may say that a person is altruistic if her utility increases with the well being of other people. Formally speaking, given the utility function $u(x_1, ..., x_N)$, its first partial derivatives, with respect to a certain allocation of resources $x_1, ..., x_N$, are strictly positive. Andreoni & Miller (2002) conducted a series of dictator game experiments, concluding that many individuals seem to have others-regarding preferences, that individuals are heterogeneous, and only a minority of subjects can be described as unconditional altruists who have a utility function that is always strictly increasing in the payoff of their opponent.

Bolton (1991) formalized the idea that individuals are not only concerned with the absolute amount of money they receive, but also with their relative amount compared to others, contextualized of an experimental two-player bargaining game. He proposed that the utility of an agent i would be defined through $U_i(x_i, x_j) = u_i(x_i, x_i/x_j)$, where the partial derivative with respect to x_i/x_j is strictly positive when $x_i < x_j$ and equal to 0, otherwise. Thus, agent i suffers if she gets less than agent j, but she does not care about what agent j gets if she is better off herself. However, while this utility function is consistent with agents behavior in those bargaining games considered by Bolton, it neither explains generosity in dictator games and kind behavior of responders in trust games and gift exchange games nor voluntary contributions in public good games, which are the common games used to test the influence of social factors.

We have assumed, so far, that utility is either monotonically increasing or monotonically decreasing in the well being of other players. Fehr & Schmidt (1999) assume that a player is altruistic towards other players if their payoffs are below an equitable standard, but she feels envy when the other players' payoffs exceed such level. Fehr and Schmidt consider the simplest utility function capturing this idea, which is

$$U_i(x_1, x_2 \ldots, x_N) = x_i - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max\{x_j - x_i, 0\} - \frac{\beta_i}{N-1} \sum_{j \neq i} \max\{x_i - x_j, 0\},$$

with $0 \leq \beta_i \leq \alpha_i$ and $\beta_i \leq 1$. This approach is consistent with generosity in dictator games and kind behavior of responders in trust games and gift exchange games, and, at the same time, with the rejection of low offers in ultimatum games. It can explain voluntary contributions in public good games, and at the same time, costly punishments of free-riders, improving Bolton's idea.

Models described so far assume that players' utility functions depend only on the final allocation of resources, but not on how different allocations were attained. For example, if I have to decide whether to accept or reject a very unequal allocation, my decision may depend on whether my opponent chose the unfair allocation deliberately, or whether he had no possibility of affecting such allocation. A possible solution is to assume that players may be of different types (e.g. altruistic and spiteful types), and that each players' preferences depend on his opponent's type.

Some models that follow this approach are Levine (1998) and its derivations. This author assumes that players differ on how altruistic they are. The model assumes that people predict other's altruism parameter (α_i) and respond with altruistic rewarding or punishment, depending on their predictions. Formally, it could be expressed through

$$U_i = x_i + \sum_j \frac{\alpha_i + \lambda_i \times \alpha_j}{1 + \lambda_i} \times x_j,$$

where α_i captures player i's altruistic behavior (being $-1 < \alpha_i < 1$), and λ_i determines player i's preference for reciprocity (being $0 \geq \lambda_i \geq 1$).

Players may possibly care about their opponents' intentions. In a pioneering article, Rabin (1993) modeled intention based reciprocity for simple two-player normal-form games. Let us denote by A_i^j the set of available payoffs to player i depending on player j's choice. Let π_i^L be the lower limit of A_i^j, and π_i^H, the upper limit of the set A_i^j. A fair payoff is defined as $\pi_i^F = (\pi_i^H + \pi_i^L)/2$. Kindness of player i towards player j is denoted by k_i^j, which is defined as 0 if $\pi_i^H = \pi_i^L$, and as $2(\pi_i^A - \pi_i^F)/(\pi_i^H - \pi_i^L)$ otherwise, being π_i^A the payoff of player i given the actual choice of player j. The utility of player i is defined as

$$U_i = x_i + \rho_i \times k_i^j \times x_i,$$

being ρ_i the individual reciprocity parameter ($\rho_i > 0$). Rabin's theory is important because it was the first contribution that precisely defined the notion of reciprocity and explored the consequences of reciprocal behavior. It is consistent with rejections in the ultimatum game, but many other equilibria exist as well, some of which are highly implausible.

Falk & Fischbacher (2006) generalize Rabin's model. They consider n-person extensive-form games and allow for the possibility of incomplete information. They show that there are parameter settings for which their model is consistent with the ultimatum game, the gift exchange game, the dictator game, and of public good and prisoners' dilemma games. Because their model contains variants of a pure intentions based reciprocity model, like Rabin (1993), and a pure inequity aversion model, like Fehr & Schmidt (1999) or Bolton & Ockenfels (2000), as special cases, it is possible to get a better fit of the data, but at a significant cost in terms of model complexity.

1.4 Affective and social decisions

It has been suggested that emotions influence decision-making, see Damasio (1994), Loewenstein & Lerner (2003), Loewenstein (2000), Camerer & Fehr (2006) and Glimcher *et al.* (2008). In this Section, we shall briefly review this literature, from the neurological triggers of emotions to mathematical models that use different representations of emotion in decision-making processes.

1.4.1 Understanding emotions

Current psychological research about emotions is divided into two main groups: those considering emotions not as independent mechanisms but as differing along a number of dimensions (such as pleasant vs. unpleasant, low vs. high arousal, and approach vs. avoidance), and those who contend that there exist distinct categories of emotions, each one with its own adaptive function which should be achieved from fundamentally distinct responses, see Keltner *et al.* (2003).

Neurological basis: limbic system. In Fig. 1.10, we show some of the areas related with emotional processing (in bold) and others related with cognitive processing (in italics). The former compose the limbic system, see Cohen (2005), which is shared with other mammals. Its function consists of regulating body temperature and drives e.g. hunger, sleep and sex. The limbic system passes information to the cortex area, the latter, where higher-level cognitive processing takes place.

It has been widely studied in neurology, mainly in relation with how social interactions are interpreted in our brains that the *Mesencephalic Dopamine Sys-*

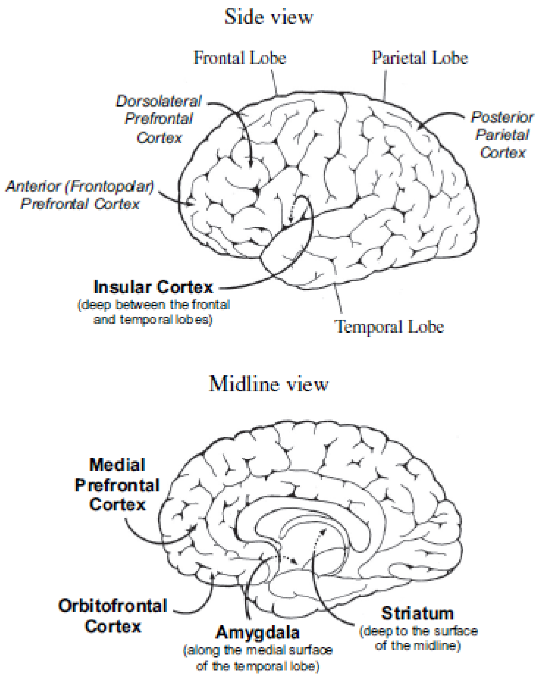


Figure 1.10: Brain areas related to decision-making, Cohen (2005).

tem plays a key role in the reward-encoding metric. Changes in the activity of the striatum, which has the dopamine receptor cells, have been demonstrated to be directly related with monetary reward or punishment, see O'Doherty (2004). Following these advances, fields like neuroeconomics emerged trying to explain how the decision-making process is reflected on the brain activity, see Glimcher *et al.* (2008) for more details.

1.4.2 Emotions and decisions

Traditionally, emotions have been excluded from the decision-making field because this was considered a cognitive process, and emotions were seen as a biasing factor in that process. Consider the following example on how emotions could determine an individual to act irrationally. Suppose a situation in which we are in a meeting to make a rational decision about ending a war. The outcome of this negotiation can turn upside down if the leader of one of the sides has just lost a member of his family before the negotiation takes place. Thus, emotions should seemingly be removed from decision-making processes, as they do not let decision-makers act rationally.

Recent research in neurology shows that acting without emotional capabilities could be very difficult or even impossible, see Damasio (1994). As Damasio remarks, one of his patients who suffered damage in his ventromedial prefrontal cortex during a surgery, scored normally in neurological and intelligence tests but could not make rationally everyday decisions anymore. In his book, Damasio details the *Somatic Marker Hypothesis* (SMH) which states that decision-making is influenced by signal markers created through bioregulatory processes. SMH is based on an assumption of a complex human decision-making process, where unconcious and concious layers interact, see Bechara *et al.* (2000).

Loewenstein & Lerner (2003) explain how emotions influence our decision-making process and argue that decisions can be based on maximizing not expected utility, but immediate and expected emotions. In Fig. 1.11 we can see the influences of two different types of emotions: expected and immediate emotions. *Expected emotions* are those which are predicted depending on future decision

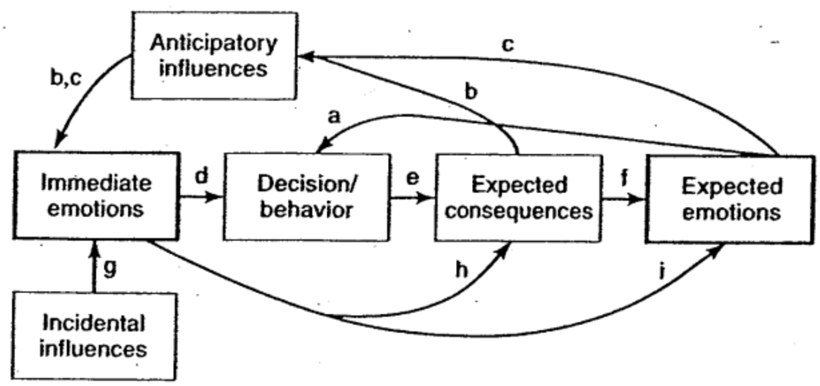


Figure 1.11: Influence of emotions in the decision-making process, see Loewenstein & Lerner (2003).

outcomes, as regret, hope, disappointment or fear. They are not emotions *per se*, at the moment of the decision-making, but rather expectations about possible emotions. Maximizing the positivity of these emotions is widely accepted by decision theorists, but assuming that this emotions encompass all the factors that decision-makers care about is controversial. *Immediate emotions* are those experienced at the precise moment of the decision, as satisfaction, happiness, sadness or anger. These emotions may provoke two types of influence (direct or indirect) on the decision-making process. Their influence could be represented as prioritizing information processing.

1.4.3 Emotional-influenced decision-making models

Throughout this Section, we introduce some of the emotional-influenced decision-making models found in the literature.

Decision Affect Theory (DAT), see Mellers *et al.* (1997), is motivated by regret and disappointment theories, and tries to capture the effect of other alternatives on the experienced benefit of an outcome. The following equation shows the formula for an emotional response to an outcome,

$$R_a = c \times \left[u_a + \sum_{a \neq b} g(u_a - u_b) \times s_b \right] + d,$$

where R_a is the emotional response; c and d, are linear coefficients in the judgement function; u_a and u_b are utilities of the obtained and unobtained outcomes; s_b is the subjective probability of event b and $g(\cdot)$ is the disappointment function, which reflects the difference between what occurred and what could have occurred under a different state of the world.

Decision Field Theory (DFT). Busemeyer *et al.* (2006) present a model that consists of a dynamic stochastic preference model which activates one alternative when a threshold value is exceeded. It is composed of weights (w_{ij}) for alternative i and consequence j, and a utility function m_j that measures consequence values. Those values are determined through

$$m_j = \sum_k n_k q_{jk},$$

where n_k represents the need k (including emotions) that are unsatisfied over time, and q_{jk} is used to associate consequence j with such need k. Thus, the utility function at time t would be:

$$U_i(t) = \sum_j w_{ij} m_j.$$

The Visceral Factors Model was introduced by Loewenstein (2000). It is based on the concept of *visceral factors*, which refers to negative emotions, drive states and feeling states that impact on people and motivate them to behave in a specific way. Loewenstein suggests that agents should maximize a utility function $u(c_i, s_j)$, where c_i is a vector of the consumption activities and s_j is a vector of individual visceral states (e.g., eating and hunger, aggression and anger), through the formula:

$$\tilde{u}(c_0, c_1, \ldots, c_T; s_0, s_1, \ldots, s_T) = u(c_0, s_0) + \sum_{t=1}^{T} \tilde{u}(c_t, s_t),$$

which represents a compromise among (1) the utility function that will be associated at time t; (2) the utility function that captures the underappreciation of future visceral factors; and (3) the individual's current utility function that captures the *hot-cold empathy gap*. With this last concept, the author refers to how difficult is to imagine positive feelings when you are in a negative feeling state, or vice versa.

State-Dependent Priorities. As described in Rázuri *et al.* (2012) and Sribhashyam & Montibeller (2012), emotional elements impact on priorities and they could be modeled using a multi-attribute utility function, where the weights assigned to the objectives depend on the emotions. Thus, the overall utility can be formally assessed at time t of the i-th alternative as follows:

$$U_i(t) = \sum_k w_k(s_t) \times u_{i,k},$$

where $w_k(s_t)$ are the emotion-dependent weights at time t for the objective k, and $u_{i,k}$ the partial utilities of alternative i on the k-th objective.

1.4.4 Computational emotional models

Artificial emotions are used in robotics for several reasons: the primary purpose, is that emotions help in facilitating believable human-robot interaction, see Cañamero & Fredslund (2001); a second reason is that artificial emotions can also provide feedback to the user, indicating the robot internal states, goals and intentions, see Breazeal (1998) and Kozima & Yano (2001); lastly, artificial emotions may act as a control mechanism, driving behavior and reflecting how the robot is affected by different factors over time, see Cañamero (1997), Michaud *et al.* (2000), and Velásquez (1998).

The theories of Roseman (1984) and Ortony *et al.* (1988) (OCC) lead the way to many computational implementations of emotions. Some of them are:

Affective Reasoner by Elliott (1992) is a computational adaptation of the OCC model which assesses the relationship between events and an agent personality, described by its goals, social standards, and preferences. The relationship is characterized in terms of a set of features called emotion-eliciting conditions. The Affective Reasoner considers social interaction in modeling emotions. The agents own knowledge of emotions and actions is used to understand the other agents emotional states, expressions, and their actions. The model does not consider the intensities of emotions. Besides, there is no proposed resolution in case of conflicting emotions.

Cathexis was proposed by Velásquez (1997) to generate artificial emotions and focus on their influence on the behavior of intelligent agents. Cathexis introduces the concept of "temperament", which determines how an agent would experience different emotions. Cathexis models six basic emotions: anger, fear, distress/sadness, enjoyment/happiness, disgust and surprise. However, most of the variables to calculate them are hard-coded inside Cathexis, which makes the system inflexible.

FLAME was designed by El-Nasr *et al.* (2000). It is a computational implementation of emotions based on fuzzy logic rules used to map the impact of events on goals into emotional intensities. FLAME consists of three components: emotional, learning and decision-making. The agent first perceives external events from the environment, and evaluates them using the emotional component, which first determines which goal is affected by the event, and then determines the desirability of that event having an impact over the agent goals. After the evaluation, the desirability is used by the appraisal process to update the emotional state of the agent. FLAME is based on a combination of the OCC model and Roseman's to trigger emotions. Its main drawback is that the events and agents' goals have to be hard coded in rules. This makes FLAME domain-dependent and inflexible.

Emile This model uses classical planning methods to evaluate the emotional impact of events on plans and goals, to model and predict the emotional state of other agents, and alter its behavior accordingly, see Gratch (2000). The core of classical planning methods is to detect and resolve threats that prevent an agent from obtaining its goals.

TAME is a very computational expensive emotional model, which combines emotions, mood, personality traits and attitudes, aiming at integrating the entire affect-related space into a single model with explicitly defined interactions, see Moshkina (2011).

Expressing Emotions on robotic agents is generally not life-like, due to limitations of mechatronics design and control. The most common facial components used are mouth (lips), cheeks, eyes, eyebrows and forehead. Most robot faces express emotions in accordance with Ekman & Friesen (1977). As an example, Kismet, see Breazeal & Scassellati (1999), has controllable eyebrows, ears, eyeballs, eyelids, a mouth with two lips and a pan/tilt neck, see Fig. 1.4(b). In our case, AiSoy1 is capable of moving its eyebrows, eyelids and neck apart from using its mouth-like led matrix and its RGB led in the middle of its body to express the predominant emotion, see Fig. 1.5.

1.4.5 Social behavior

Self-interest has been considered, for a long time, as the sole motivation of decision-makers. Indeed, in economics, this approach has served very well in domains such as competitive experimental markets. However, in strategic interactions, where individual's choices affect other individual's payoffs, this approach fails to predict correctly, see Camerer & Fehr (2006), and a social preference model approach arises, see Fehr (2008). The term *social preferences* is mainly used as a characteristic of an individual's behavior, remarking whether the individual cares positively or negatively about other's payoffs.

Some experiments, see Camerer (2003) and Fehr & Schmidt (1999), have demonstrated that in fact, a substantial percentage of people are motivated by the well-being of others, and there is an important heterogeneity in social preferences. However, the presence of social preferences does not mean that individuals make social decisions no matter what costs they must deal with. Otherwise, they should be considered as an important element in individual utility function, see Fehr (2008).

The striatum appears to play a central role in social decisions. At the beginning of this Section, we mentioned that it is involved in the reward-encoding system of the brain. Sanfey & Dorris (2008) show the important function that the striatum plays in iterated games like trust or prisoner's dilemma's games, tracking a social partner's decision to whether reciprocate or not cooperate in those games.

Some studies have demonstrated that negative emotions, such as sadness and disgust, lead to higher rejection rates of unfair offers when playing ultimatum games, see Harle & Sanfey (2007) and Damasio *et al.* (2000). Fiori *et al.* (2013) study how, apart from emotional states, personality traits may affect the performance of individuals playing ultimatum games. In such work, authors conclude that positive emotions imply a higher acceptance rate, and negative emotions higher amount of money offered, while introvert, conscientious and honest human responders were more likely to accept unfair offers. These emotional reactions have been seen as a mechanism through which to avoid inequity, and may have evolved to support mutual reciprocity, making reputation important, and punish those who are seeking to take advantage of others, see Nowak *et al.* (2000).

As we have seen, in game theory players choose strategies that maximize their expected utility, given their beliefs about what other players are likely to do. The question then is how those beliefs are formed. *Behavioral game the-*

ory, see Camerer (2003), is aimed at showing how players actually make their decisions in realistic situations. Many experiments have been done, see Camerer (2008), that suggest that traditional game theory sometimes explains behavior surprisingly well, but sometimes results are far from being accurate. Behavioral game theory aims at generalizing game theoretic ideas in order to explain experimentally-observed violations by incorporating bounds on rationality in a formal way. Rogers *et al.* (2009) combines different approaches as Cognitive Hierarchy theories (CH), which limit the strategic thinking of the players, and Quantal Response Equilibrium theories (QRE) which assumes that each player may make mistakes while they compute their beliefs, to explain within a unique model why behavior is sometimes far from equilibrium and remarkably close in others.

1.5 Problems posed

Through this Chapter we have seen the increasing interest of Humankind in building machines that help us in our daily tasks, and which are the steps that scientists are undertaking towards that goal. We have reviewed some of the mathematical methods employed to make robotic agents interact among them and include social and emotional factors as key components for designing believable agents.

Throughout this thesis, we shall aim at designing social and emotional models to be implemented in non-expensive robotic platforms, and at solving certain questions related with the Adversarial Risk Analysis framework as well as the Intelligent and the Multi-Agent Systems fields. As pointed out in Section 1.1, the emotional model implementated on the AiSoy1 robot turned out to be too emotionally unstable, being this and its low-cost the main reasons to choose this platform as the one to implement our designed models in.

As a context for this work, suppose we have a finite set of robots that can make decisions within a finite set of actions. Such robots have sensors that allow them to obtain information about the environment that surrounds them and interact with it. At the same time, in our scenario, we have a finite set of human participants that can also make actions defined in a finite set of actions. Those humans may interact with the robots, and with the environment too. Within such scenario, we have developed several decision-making models that are expected to solve different challenges concerning different scientific fields like how the Adversarial Risk Analysis framework may be extended to the robotic field; how it performs compared with game theoretic approaches within competitive and cooperative scenarios related to the Multi-Agent Systems; despite the amount of emotional models designed until now, none of them may be applied on non-expensive platforms due to the complexity increase they required, so that a model has been designed expecting to fill that gap in the literature. We aimed at demonstrating that implementing these models in commercial robotic agents would be useful in real situations with children.

The rest of this thesis is structured as follows: in Chapter 2, an Adversarial Risk Analysis model for an autonomous decision agent is detailed. Through Chapter 3, we extend this model to situations in which the agents cooperate and compete in multi-agent systems, exploring the social needs of our robotic agent, and how it handles interactions with several agents (both humans and robotic ones). Chapter 4 consists on extending the basic model to a case of affective decision-making, including an emotional model that allows the agent to simulate emotions depending on the evolution of the environment and the behavior of the users. Chapter 5 provides some conclusions and future work.

Chapter 2

An Adversarial Risk Analysis Model for an Autonomous Decision Agent

2.1 Introduction

One of the main current challenges in robotics is the development of models that may be implemented in non-expensive platforms, with its entailed computational limitations. Our first model aims at supporting the decision-making of an imperfect autonomous agent who is able to make decisions facing a user, based on the previously introduced framework of Adversarial Risk Analysis (ARA), see Section 1.3.5.

This model has been implemented with an AiSoy1, endowed with several sensors to infer the user's actions and environment's states, see Section 1.1. As we shall see, its processing capacities leads us to develop simple, yet rigorous, models. Some computational experience is included at the end of the Chapter, whose structure is as follows: in Section 2.2, we describe the basic elements and participants in our framework. Section 2.3 defines the forecasting and preference models and the relevant expected utility maximization problem. The implementation of our model is described in Section 2.4, including a detailed explanation of the forecasting and multiobjective preference models. It also includes conclusions about the simulations. An implementation with an AiSoy1 robot is provided in Section 2.6. Finally, in Section 2.7, we end up with a discussion.

2.2 Basic framework

We start by introducing the basic elements in our model. We aim at designing an agent A whose activities we want to regulate and plan. There is another participant B, the user, which interacts with A. The activities of both A and B take place within an environment E. As a motivating example, suppose that we aim at designing a robot A which will interact with a child B, within a room E.

A makes decisions within a finite set $\mathcal{A} = \{a_1, \ldots, a_m\}$, which possibly includes a *do nothing* action. B makes decisions within a set $\mathcal{B} = \{b_1, \ldots, b_n\}$, which also includes a *do nothing* action. $\mathcal{B}$ will be as complete as possible, while simplifying all feasible alternatives down to a finite number. The environment E changes with the user's actions, adopting a state e within a finite set $\mathcal{E}$.

The agent faces this changing environment since it affects its own behavior. A has q sensors providing synchronous readings about the external environment. Each sensor reading is attached to a time t, being the vector of sensor readings $s_t = (s_t^1, \ldots, s_t^q)$. The agent infers the environmental state e_t based on a, possibly probabilistic, transformation function f, so that

$$\hat{e}_t = f(s_t).$$

A employs the sensor readings to infer what the user has done, based on a, possibly probabilistic, function g which provides estimates of the user's actions,

$$\hat{b}_t = g(s_t).$$

Given the nature of our application, we shall assume that sensor readings are captured without error, except for sound and video. The processing of sensor information entails probabilistic manipulation within a limited processing environment. Hence we deal with and imperfect agent, see Guy *et al.* (2013), which deals with another decision-maker.

We design our agent by planning its activities according to the basic loop in Fig. 2.1, which is open to interventions, see West & Harrison (1997), if an exception occurs. Indeed, at the begining of each iteration, sensors s_t are periodically read in order to infer the state e_t and the user's action b_t. Once the robot has detected a change within the environment or an action from the user, the next step is to update the forecasting model with the information just received. Using the recently updated forecasting model, it chooses its next action by maximizing expected utility and the clock is updated.

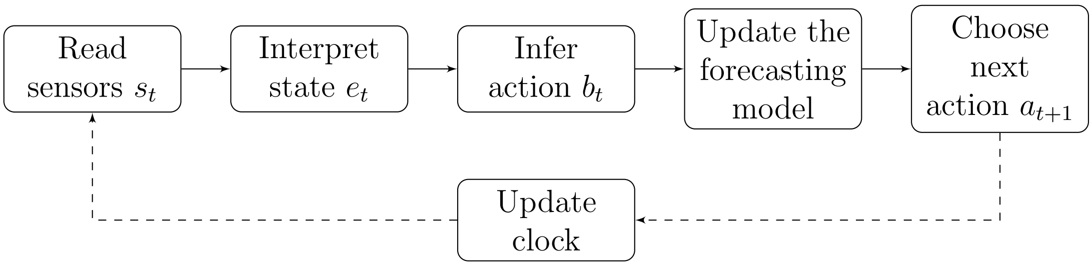


Figure 2.1: Basic agent loop.

2.3 Adversarial Risk Analysis decision model

Essentially, we shall plan our agent's activities over time within the decision analytic framework, see Clemen & Reilly (2004), including models to forecast the user behavior and the evolution of the environment. We describe, in turn, the forecasting model, which incorporates the ARA elements; the preference model; and, finally, the corresponding optimization problem.

2.3.1 Forecasting models

The agent maintains a forecasting model which suggests with which probabilities will the user act and the environment react, given the past history of the agent's actions, the user's actions and the evolution of the environment and its own action a_t.

We describe the general structure of the model. Assume that, for computational reasons, we limit the agent's memory to two instant times. We shall just forecast one period ahead. We are interested in computing

$$p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})), \tag{2.1}$$

which describes the dependence of the environment e_t and the user action b_t on the agent action and the past two events. (2.1) may be decomposed through

$$\begin{gathered} p(e_t|b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}))\times \\ p(b_t|a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})). \end{gathered} \tag{2.2}$$

We describe the first factor, which we call the *environment model*. We assume that the environment is fully under control by the user. In our motivating example, she controls the light, the temperature and other features of the room. Moreover, she may plug in the robot to charge its battery, and so on. In general, only the latest of the user's actions will trigger the evolution of the environment. Thus, we shall assume that

$$p(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = p(e_t \mid b_t, e_{t-1}, e_{t-2}).$$

Regarding the second factor in (2.2), we shall consider that the user has her own behavior evolution, that might be affected by how she reacts to the agent's actions, thus incorporating the ARA principle, as the agent forecasts how the user will react to its actions, and will use it in its decision-making, as described below. Thus, we assume that

$$p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = p(b_t \mid a_t, b_{t-1}, b_{t-2}). \tag{2.3}$$

The agent will maintain two models, M_i with $i \in \{1, 2\}$, in connection with (2.3). The first one, M_1, describes the evolution of the user by herself, assuming that she is in control of the whole environment and is not affected by the agent's actions.

We call it the *user model* and describe it through

$$p(b_t \mid b_{t-1}, b_{t-2}).$$

The other one, M_2, refers to the user's reactions to the agent's actions. We describe it through

$$p(b_t \mid a_t).$$

We call it the *classical conditioning model*, with the agent possibly conditioning the user.

We combine both models to recover (2.3). We view the problem as one of model averaging, see Hoeting *et al.* (1999) and Clyde & George (2004). In this case,

$$p(b_t \mid a_t, b_{t-1}, b_{t-2}) = p(M_1)\ p(b_t \mid b_{t-1}, b_{t-2}) + p(M_2)\ p(b_t \mid a_t),$$

where $p(M_i)$ denotes the probability that the agent gives to model M_i, with $p(M_1) + p(M_2) = 1,\ p(M_i) \geq 0$. These probabilities, essentially, capture how reactive to the agent's actions the user is.

Extensions to forecasting several steps ahead follow a similar path. For example, for two steps ahead, we have

$$\begin{aligned}
&p((e_{t+1}, b_{t+1}), (e_t, b_t) \mid a_{t+1}, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \\
&= p((e_{t+1}, b_{t+1}) \mid a_{t+1}, (e_t, a_t, b_t), (e_{t-1}, a_{t-1}, b_{t-1}))\times \\
&\times\ p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \\
&= \Big[p(e_{t+1} \mid b_{t+1}, e_t, e_{t-1}) \times p(b_{t+1} \mid a_{t+1}, b_t, b_{t-1}) \Big] \times \\
&\times \Big[p(e_t \mid b_t, e_{t-1}, e_{t-2}) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}) \Big].
\end{aligned}$$

In Section 2.4, we describe how do we learn about various model components when data is available within a particular implementation.

2.3.2 Preference model

We sketch now the preference model. Assume that the agent faces multiple consequences $c = (c_1, c_2, \dots, c_l)$. At each instant t, these will depend on its action a_t, the user's action b_t and the state e_t, occurring after a_t and b_t. Therefore, the consequences will be of the form

$$c_i(a_t, b_t, e_t), \; i = 1, \dots, l.$$

We shall assume that they are evaluated through a multi-attribute utility function, see Clemen & Reilly (2004). Specifically, we shall adopt an additive form

$$u(c_1, c_2, \dots, c_l) = \sum_{i=1}^{l} w_i u_i(c_i),$$

with $w_i \geq 0, \; \sum_{i=1}^{l} w_i = 1$, where u_i represents the robot's i-th component utility function and w_i, the corresponding utility weight.

We shall focus on cases in which the agent's objectives are ordered hierarchically, as in the hierarchy of needs in Maslow (1943), implemented by assigning higher weights to the more important objectives. See Section 2.4 for our implementation in an specific case.

2.3.3 Expected utility

Our agent aims at maximizing the predictive expected utility. Planning $(r + 1)$ instants ahead requires computing maximum expected utility plans defined through:

$$\max_{(a_t,\dots,a_{t+r})} \psi(a_t, \dots, a_{t+r}) = \int \cdots \int \left[\sum_{i=0}^{r} u(c(a_{t+i}, b_{t+i}, e_{t+i})) \right] \times$$

$$\times \; p((e_t, b_t), \dots, (e_{t+r}, b_{t+r}) \mid (a_t, a_{t+1}, \dots, a_{t+r}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}))) db_t de_t \dots db_{t+r} de_{t+r},$$

assuming utilities to be additive over time. This could be solved through dynamic programming, through Bellman (1957)'s equation, which, e.g., for the case of two periods ahead would be:

$$\arg\max_{a_t,a_{t+1}\in\mathcal{A}} \psi(a_t,a_{t+1}) = \arg\max_{a_t,a_{t+1}\in\mathcal{A}} \int\int\int\int \Big[u(a_t,b_t,e_t)+u(a_{t+1},b_{t+1},e_{t+1})\Big]\times$$

$$\times p((e_t,b_t)\mid a_t,(e_{t-1},a_{t-1},b_{t-1}),(e_{t-2},a_{t-2},b_{t-2}))\times$$

$$\times p((e_{t+1},b_{t+1})\mid a_{t+1},(e_t,a_t,b_t),(e_{t-1},a_{t-1},b_{t-1}))db_t de_t db_{t+1} de_{t+1} =$$

$$= \arg\max_{a_t\in\mathcal{A}} \int\int \Big[u(a_t,b_t,e_t)+V_{t+1}((e_{t-1},b_{t-1}),(e_t,b_t))\Big]\times$$

$$\times p((e_t,b_t)\mid a_t,(e_{t-1},a_{t-1},b_{t-1}),(e_{t-2},a_{t-2},b_{t-2}))db_t de_t.$$

The equation in this case is

$$V_t((e_{t-2},b_{t-2}),(e_{t-1},b_{t-1})) = \max_{a_t\in\mathcal{A}} \int\int \Big[u(c(a_t,b_t,e_t))+V_{t+1}((e_{t-1},b_{t-1}),(e_t,b_t))\Big]\times$$

$$\times p((e_t,b_t)\mid a_t,(e_{t-1},a_{t-1},b_{t-1}),(e_{t-2},a_{t-2},b_{t-2}))db_t de_t,$$

where $V_t((e_{t-2},b_{t-2}),(e_{t-1},b_{t-1}))$ would be the value function at time t, assuming that the past history that defined the state at that time would be (e_{t-2},b_{t-2}), (e_{t-1},b_{t-1}). The final condition would be:

$$V_{t+1}((e_{t-1},b_{t-1}),(e_t,b_t)) = \max_{a_{t+1}\in\mathcal{A}} \int\int u(c(a_{t+1},b_{t+1},e_{t+1}))\times$$

$$\times p((e_{t+1},b_{t+1})\mid a_{t+1},(e_t,a_t,b_t),(e_{t-1},a_{t-1},b_{t-1}))db_{t+1} de_{t+1}.$$

If planning several instants ahead turns out to be very expensive computationally, we could plan just one period ahead. In this case, we would aim at solving

$$\max_{a_t\in\mathcal{A}} \psi(a_t) = \int\int u(c(a_t,b_t,e_t))p(e_t|b_t,e_{t-1},e_{t-2})\ p(b_t|a_t,b_{t-1},b_{t-2})db_t, de_t.$$

We may mitigate the myopia of this approach by adding a term penalizing deviations from some ideal agent consequences, as in Ríos Insua & Salewicz (1995).

In this case, the utility would have the form $u(c) - \rho(c, c^*)$ where ρ is a distance and c^* is an ideal consequence value.

Agents operating in this way may end up being too predictable. We may reduce such effect by choosing the next action in a randomized way, with probabilities proportional to predictive expected utilities assuming, without loss of generality, that they are all non-negative, that is

$$P(a_t) \propto \psi(a_t),$$

where $P(a_t)$ is the probability of choosing a_t. See Patè-Cornell & Guikema (2002) for a justification of such approach.

2.4 Implementation

The above procedures have been implemented within the AiSoy1 robot environment. The details of the model implemented are described next.

2.4.1 Basic elements

The robot's alternatives in $\mathcal{A}$ include actions for calling the user's attention, several options to interact with the user and a *do nothing* action, reflected in Fig. 2.2. The set of actions have been designed according to the expected possible interactive situations among the user and the robotic agent. The robot has 15 alternatives, $\mathcal{A} = \{a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}\}$ = {*ask for help, salute, ask for charging, complain, play, speak, ask for playing, ask for shutting down, tell jokes, tell events, obey, flatter, offend, apologize, do nothing*}, which are described next. Some actions would be only considered if some conditions are met, as specified below. We classify these alternatives as follows:

Call for user's attention

Ask for help: Used only when the robot feels insecure, see Section 2.4.3. The agent calls the user and tells her about it.

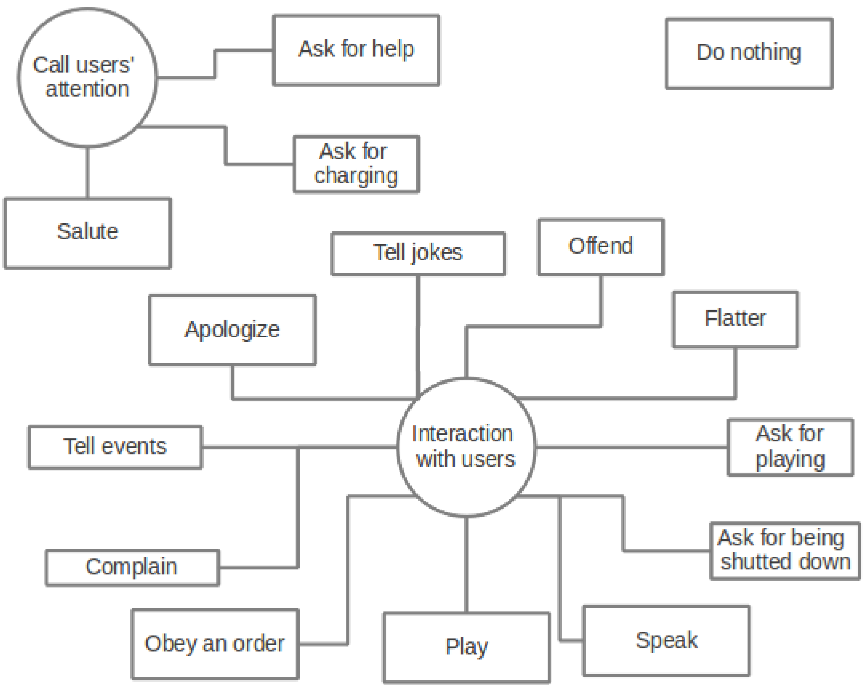


Figure 2.2: Robot actions.

Salute: Used when a user appears within the scene. The agent salutes the user.

Ask for charging: Used only when the battery level is under 20 %. The agent asks the user to be charged.

Interaction with users

Complain: Used only when the utility of the perceived consequences, see Section 2.4.3, is below a certain threshold. Depending on such utility, the agent will *warn* the user when it is below 0.4, *alert* the user that something is not going right, if it goes below 0.3, and finally, *cry*, if the utility lies under 0.2.

Play: Once a game has been started the agent participates as required.

Speak: The agent participates into a conversation with the user.

Ask for playing: The agent suggests the user to play a game.

Ask for shutting down: Used only when the utility of the consequences is under 0.15. The agent asks the user to be shutted down.

Tell jokes: The agent wants to entertain the user telling her jokes.

Tell events: The agent wants to explain the user who it is, what happened recently, etc.

Obey: Used only when the user ordered something to the agent, e.g., turn your volume up, stay calm for a while, etc.

Flatter: The agent wants to express its gratitude towards its user.

Offend: The agent wants to express its ingratitude towards its user.

Apologize: Used only if the agent's last action was *offend*.

Other actions

Do nothing: The agent prefers the *status quo*.

On the user's side, set $\mathcal{B}$, the robot is able to detect several user's actions, some of them in a probabilistic way. Indeed, the robot detects three types of actions based on the impact they made on its preference model: affective, aggressive, and interacting actions, see Fig. 2.3. The robot will also detect whether the user took *no action*. This totals $n = 16$ actions with $\mathcal{B} = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, b_{16}\}$ = {*recharge, stroke, flatter, apologize, attack, offend, move, blind, shout, discharge, speak, ignore, order, play, update, do nothing*}.

The detection of some actions is based on simple deterministic rules. For example, the action *move* is interpreted through a detection in a touch sensor and a variation in the inclination sensor over the last 2 instants, given the memory limitation of the robot. An *instant* may be defined as the period of time required to complete one iteration through the basic loop in Fig. 2.1. We have fixed the considered number of instants to 2 based on trial and error. Some other actions are detected according to probabilistic rules, like those involving voice recognition and processing, see Section 1.1 where the ASR component is described. We provide the details of how the user's actions are detected, dividing them into

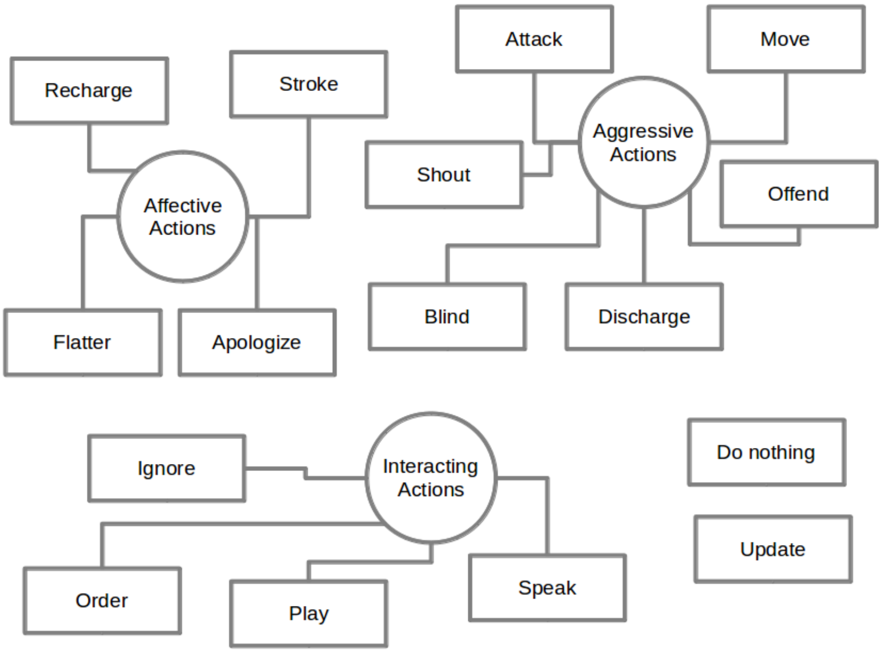


Figure 2.3: User's actions.

deterministic and probabilistic rules. Just one action is detected at a single time. If probabilistic rules are involved, the most likely action is taken as the detected one. In the case of a conflict between deterministic and probabilistic rules, the deterministic rule will determine the inferred action.

Deterministic rules

Affective actions

Recharge: The user is charging the robot. It is inferred when the battery level is $< 100\ \%$, the power supply wire is plugged in and the power supply status is different from the last power supply reading.

Stroke: The user has stroked the agent. Such action is inferred when, while in vertical position, there is no change in the inclination sensor at the next *2 instants* and it is touched during them.

Aggressive actions

Discharge: The user has recently unplugged the robot. It is inferred when the power supply wire is not plugged in and the power supply status is different from the last power supply reading.

Attack: The user has pushed or hit the robot. An attack is inferred when there are changes in the inclination during the following *2 instants* or the robot is not in vertical position.

Move: The user is moving the robot from one place to another. Such action is inferred when there are changes in the inclination during the following *2 instants* and contact is detected during them.

Blind: The user has just turned the light off or has placed her hand in front of the agent's eyes. Such action is inferred when there is no light detected at the following *2 instants* and light was detected at the last *2 instants*.

Shout: The user has just made a loud noise. It is inferred whenever the noise level is suddenly increased.

Other actions

Update: The user has updated the agent's software. Such action is inferred whenever the agent detects a difference in the software version when rebooted.

Do nothing: The user did not perform any of the recognizable actions. Such action is inferred when the user is detected within the scene but she does not perform any of the defined actions.

Probabilistic rules

Affective actions

Flatter: The user has said something nice to the agent. It is inferred if there are words detected within a specific set [rewards, compliments] and the user is present or the name of the robot is mentioned.

Aggressive actions

Offend: The user has said something inappropiate to the agent. It occurs whenever there are words detected within a specific set [insults, threats] and the presence of the user or the name of the robot is detected.

Interactive actions

Speak: The user has started a conversation with the robot. Such action is inferred if the agent detects the presence of the user or the name of the robot and the words detected belong to a specific set [standard-issues, grettings].

Play: The user has asked the robot to play some game or they are already playing a game together. It is inferred when the agent has detected the presence of the user or the name of the robot and the user has asked for a game using words within a specific set [games].

Order: The user has ordered the agent to perform some task, like turn its volume down, take a picture, etc. It is inferred whenever the agent detects the presence of the user or the name of the robot and there are words detected within a specific set [orders].

Ignore: The user is ignoring the agent. Such action is inferred whenever the user is within the scene and there is no response from her on the following *2 instants*.

As an example of a probabilistic rule, like b_3=*flatter*, the detection is based on the ASR component through which the robot will detect various words with certain probabilities. The word which maximizes such probabilities, is considered as identified, and it is sent to the ChatScript module to be interpreted, see Section 1.1. If the interpretation is related with the rewards or compliments sets, (for example the compliment “nice robot”), the robot will interpret this as being flattered.

2.4.2 Forecasting model

We describe now how we have implemented the relevant forecasting models. D_t will designate the data available until time t. When required, posteriors refer to data available until such time.

The classical conditioning model

This model forecasts the user's action b_t based on the agent action a_t. We shall use a matrix-beta model for such purpose, Ríos Insua *et al.* (2012). For each a_j, the prior distribution will be Dirichlet with parameters $\beta_{ij} \geq 0, i \in \{1, ..., n\}$, so that

$$p(b_t \mid a_t = a_j) \sim Dir(\beta_{1j}, \ldots, \beta_{nj}),\ \ b_t \in \{b_1, b_2, ..., b_n\}.$$

Now, if h^t_{ij} designates the number of occurrences of the user doing b_i, when the robot has made a_j until time t, the posterior distribution will be

$$p(b_t \mid a_t = a_j, D_t) \sim Dir(\beta_{1j} + h^t_{1j}, \ldots, \beta_{nj} + h^t_{nj}),\ \ b_t \in \{b_1, b_2, ..., b_n\}.$$

When necessary, we may summarize them through the expectations

$$\hat{p}_{ij} = E[p(b_t) = b_i | a_t = a_j, D_t] = \frac{\beta_{ij} + h^t_{ij}}{\sum_i (\beta_{ij} + h^t_{ij})},\ \ i \in \{1, 2, ..., n\}, j \in \{1, 2, ..., m\}.$$

The required data will be stored in the matrix structure

$$\begin{pmatrix} \beta^t_{11} = \beta_{11} + h^t_{11} & \cdots & \beta^t_{1m} = \beta_{1m} + h^t_{1m} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \beta^t_{n1} = \beta_{n1} + h^t_{n1} & \cdots & \beta^t_{nm} = \beta_{nm} + h^t_{nm} \\ \beta^t_{(n+1)1} = \sum_{i=1}^n (\beta_{i1} + h^t_{i1}) & \cdots & \beta^t_{(n+1)m} = \sum_{i=1}^n (\beta_{im} + h^t_{im}) \end{pmatrix}$$

whose last row accumulates the sum of row values for each column. At each relevant time t, we shall increment the corresponding ij-th element of the matrix and the corresponding element of the last row: if the action sequence is $a_{t+1} = a_j,\ b_{t+1} = b_i$, we shall update $\beta^{t+1}_{ij} = \beta^t_{ij} + 1$ and $\beta^{t+1}_{(n+1)j} = \beta^t_{(n+1)j} + 1$, with the remaining entries satisfying $\beta_{ij}^{t+1} = \beta_{ij}^t$.

Since we expect lots of data, the terms β_{ij} will not matter that much after a while. Thus, we shall use the following prior assessment: if a couple of actions $a_t = a_j$ and $b_t = b_i$ is compatible, we shall make $\beta_{ij} = 1$; otherwise, we shall make $\beta_{ij} = 0$. For example, action a_3= *ask for charging* is incompatible with action b_{10} = *discharge* as the bot is asking for charge because it is not plugged in.

The user's model

We provide now our forecasting model for the current user's action b_t based on what the user, b_{t-1} and b_{t-2}, has done two time steps before. As in the previous one, we use a matrix-beta model. For $i, j \in \{1, 2, \ldots, n\}$, we have a priori

$$p(b_t \mid b_{t-1} = b_j, b_{t-2} = b_i) \sim Dir(\rho_{ij1}, \ldots, \rho_{ijn}),\ \ b_t \in \{b_1, b_2, ..., b_n\}.$$

If h_{ijk}^t designates the number of occurrences in which the user performed $b_t = b_k$ after having implemented $b_{t-1} = b_j$ and $b_{t-2} = b_i$, we have that the posterior distribution is

$$p(b_t \mid b_{t-1} = j, b_{t-2} = i, D_t) \sim Dir(\rho_{ij1} + h_{ij1}^t, \ldots, \rho_{ijn} + h_{ijn}^t),\ \ b_t \in \{b_1, b_2, ..., b_n\},$$

which we may summarize, when needed, through the expectations

$$\hat{p}_{ijk} = E[p(b_t) = b_k | b_{t-1} = b_j, b_{t-2} = b_i, D_t] = \frac{\rho_{ijk} + h_{ijk}^t}{\sum_k (\rho_{ijk} + h_{ijk}^t)},\ \ k \in \{1, 2, ..., n\}.$$

The data structure used to store the required information will consist of a three-dimensional matrix,

$$
\begin{array}{ccc}
 & \rho^t_{n11} = \rho_{n11} + h^t_{n11} & \rho^t_{nn1} = \rho_{nn1} + h^t_{nn1} \\
\rho^t_{111} = \rho_{111} + h^t_{111} & \rho^t_{1n1} = \rho_{1n1} + h^t_{1n1} & \\
 & & \rho^t_{nnn} = \rho_{nnn} + h^t_{nnn} \\
\rho^t_{11n} = \rho_{11n} + h^t_{11n} & \rho^t_{1nn} = \rho_{1nn} + h^t_{1nn} & \rho^t_{nn(n+1)} = \sum_k (\rho_{nnk} + h^t_{nnk}) \\
\rho^t_{11(n+1)} = \sum_k (\rho_{11k} + h^t_{11k}) & \rho^t_{1n(n+1)} = \sum_k (\rho_{1nk} + h^t_{1nk}) &
\end{array}
$$

whose last row accumulates the sum of values for each column. As before, at each time instant, we update the corresponding ijk-th element and the corresponding last row of the cube. The ρ_{ijk}'s elements are assessed as above.

Model averaging

We describe now how model averaging and updating takes place within our model. First, recall that we shall use

$$p(b_t \mid a_t, b_{t-1}, b_{t-2}, D_t) =$$

$$= p(M_1 \mid D_t) p(b_t \mid b_{t-1}, b_{t-2}, D_t) + p(M_2 \mid D_t) p(b_t \mid a_t, D_t),$$

with

$$p(M_i \mid D_t) = \frac{p(D_t \mid M_i) p(M_i)}{\sum_{i=1}^{2} p(D_t \mid M_i) p(M_i)}, \ i = 1, 2.$$

Under the assumption $p(M_1) = p(M_2) = \frac{1}{2}$,

$$p(M_i \mid D_t) = \frac{p(D_t \mid M_i)}{\sum_{i=1}^{2} p(D_t \mid M_i)}, \ i = 1, 2,$$

with

$$p(D_t \mid M_i) = \int p(D_t \mid \theta_i, M_i) p(\theta_i \mid M_i) d\theta_i,$$

where θ_i is the vector of parameters of model M_i, $p(\theta_i \mid M_i)$ is the prior density of θ_i under model M_i and $p(D_t \mid \theta_i, M_i)$ is the likelihood.

We provide now the computations for our models:

- M_1. We have

$$p(D_t \mid M_1) = \int \cdots \int \prod_i^n \left(\prod_j^n \left(\prod_k^n p_{ijk}{}^{h_{ijk}^t} \right) \right) \times$$
$$\times \prod_i^n \left[\prod_j^n \left(\frac{\Gamma(\sum_k \rho_{ijk})}{\prod_k^n \Gamma(\rho_{ijk})} \left(\prod_k^n p_{ijk}^{\rho_{ijk}-1} \right) \right) \right] dp_{ijk} =$$
$$= \left(\prod_i^n \left(\prod_j^n \frac{\Gamma(\sum_k \rho_{ijk})}{\prod_k^n \Gamma(\rho_{ijk})} \right) \right) \left(\prod_i^n \left(\prod_j^n \frac{\prod_k^n \Gamma(\rho_{ijk} + h_{ijk}^t)}{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^t))} \right) \right) \times$$
$$\times \int \cdots \int \prod_i^n \left[\prod_j^n \left(\frac{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^t))}{\prod_k^n \Gamma(\rho_{ijk} + h_{ijk}^t)} \left(\prod_k^n p_{ijk}^{\rho_{ijk}+h_{ijk}^t-1} \right) \right) \right] dp_{ijk}.$$

The value of this last integral is 1, so that

$$p(D_t \mid M_1) = \prod_i^n \left(\prod_j^n \left(\prod_k^n \frac{\Gamma(\rho_{ijk} + h_{ijk}^t)}{\Gamma(\rho_{ijk})} \times \frac{\Gamma(\sum_k \rho_{ijk})}{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^t))} \right) \right).$$

Assume that the $(t+1)$ transition is from i^*, j^* to k^*. Then, $h_{ijk}^{t*} = h_{ijk}^t + 1$ if $i = i^*$, $j = j^*$ and $k = k^*$, and $h_{ijk}^{t*} = h_{ijk}^t$ otherwise. Therefore,

$$p(D_{t+1} \mid M_1) = \prod_i^n \left(\prod_j^n \left(\prod_k^n \frac{\Gamma(\rho_{ijk} + h_{ijk}^{t*})}{\Gamma(\rho_{ijk})} \times \frac{\Gamma(\sum_k \rho_{ijk})}{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^{t*}))} \right) \right) =$$

$$= p(D_t \mid M_1) \times \left[\frac{\Gamma(\rho_{ijk} + h_{ijk}^{t*} + 1)}{\Gamma(\rho_{ijk} + h_{ijk}^{t*})} \times \frac{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^{t*}))}{\Gamma(\sum_k (\rho_{ijk} + h_{ijk}^{t*}) + 1)} \right].$$

As $\Gamma(z+1) = z\,\Gamma(z)$, we simplify it to:

$$p(D_{t+1} \mid M_1) = p(D_t \mid M_1) \times \frac{(\rho_{ijk} + h_{ijk}^{t*})}{\sum_k (\rho_{ijk} + h_{ijk}^{t*})} = p(D_t \mid M_1) \times \frac{\rho_{ijk}^t}{\rho_{ij(n+1)}^t}.$$

- M_2. We have

$$p(D_t \mid M_2) = \int \ldots \int \prod_j^m \left(\prod_i^n p_{ij}{}^{h_{ij}^t} \right) \prod_j^m \left[\left(\frac{\Gamma(\sum_i \beta_{ij})}{\prod_i^n \Gamma(\beta_{ij})} \right) \left(\prod_i^n p_{ij}^{\beta_{ij}-1} \right) \right] dp_{ij} =$$

$$= \left(\prod_j^m \frac{\Gamma(\sum_i \beta_{ij})}{\prod_i^n \Gamma(\beta_{ij})} \right) \left(\prod_j^m \frac{\prod_i^n \Gamma(\beta_{ij} + h_{ij}^t)}{\Gamma(\sum_i (\beta_{ij} + h_{ij}^t))} \right) \times$$

$$\times \int \ldots \int \prod_j^m \left(\left(\frac{\Gamma(\sum_i (\beta_{ij} + h_{ij}^t))}{\prod_i^n \Gamma(\beta_{ij} + h_{ij}^t)} \right) \left(\prod_i^n p_{ij}^{\beta_{ij} + h_{ij}^t - 1} \right) \right) dp_{ij}.$$

The last integral above is 1, so that

$$p(D_t \mid M_2) = \prod_j^m \left(\prod_i^n \frac{\Gamma(\beta_{ij} + h_{ij}^t)}{\Gamma(\beta_{ij})} \times \frac{\Gamma(\sum_i \beta_{ij})}{\Gamma(\sum_i (\beta_{ij} + h_{ij}^t))} \right).$$

If the $(t+1)$ transition is from i^* to j^* then $h_{ij}^{t*} = h_{ij}^t + 1$ if $i = i^*$ and $j = j^*$, and $h_{ij}^{t*} = h_{ij}^t$, otherwise. Therefore,

$$p(D_{t+1} \mid M_2) = \prod_j^m \left(\prod_i^n \frac{\Gamma(\beta_{ij} + h_{ij}^{t*})}{\Gamma(\beta_{ij})} \times \frac{\Gamma(\sum_i \beta_{ij})}{\Gamma(\sum_i (\beta_{ij} + h_{ij}^{t*}))} \right) =$$

$$= p(D_t \mid M_2) \times \left[\frac{\Gamma(\beta_{ij} + h_{ij}^{t*} + 1)}{\Gamma(\beta_{ij} + h_{ij}^{t*})} \times \frac{\Gamma(\sum_i (\beta_{ij} + h_{ij}^{t*}))}{\Gamma(\sum_i (\beta_{ij} + h_{ij}^{t*}) + 1)} \right],$$

which, as before, we simplify to

$$p(D_{t+1} \mid M_2) = p(D_t \mid M_2) \times \frac{(\beta_{ij} + h_{ij}^{t*})}{\sum_i (\beta_{ij} + h_{ij}^{t*})} = p(D_t \mid M_2) \times \frac{\beta_{ij}^t}{\beta_{(n+1)j}^t}.$$

Note that

$$p(M_1 \mid D_{t+1}) = \frac{p(D_{t+1} \mid M_1)}{p(D_{t+1} \mid M_1) + p(D_{t+1} \mid M_2)} = \frac{1}{1 + \frac{p(D_{t+1} \mid M_2)}{p(D_{t+1} \mid M_1)}}.$$

Now

$$\frac{p(D_{t+1} \mid M_2)}{p(D_{t+1} \mid M_1)} = \frac{p(D_t \mid M_2)}{p(D_t \mid M_1)} \times \frac{\beta_{ij}^t \ \rho_{ij(n+1)}^t}{\beta_{(n+1)j}^t \ \rho_{ijk}^t},$$

which, if we write $g_1^{t+1} = \frac{p(D_{t+1}|M_2)}{p(D_{t+1}|M_1)}$, may be rewritten as

$$g_1^{t+1} = g_1^t \times \frac{\beta_{ij}^t \ \rho_{ij(n+1)}^t}{\beta_{(n+1)j}^t \ \rho_{ijk}^t}.$$

Then,

$$p(M_1 \mid D_{t+1}) = \frac{1}{1 + g_1^{t+1}},$$

and

$$p(M_2 \mid D_{t+1}) = 1 - p(M_1 \mid D_{t+1}). \tag{2.4}$$

The environment model

We describe now the environment model. Based on the set of identifiable actions that our robotic platform may infer from its sensors, we consider seven environmental variables $e_t = (e_t^1, e_t^2, e_t^3, e_t^4, e_t^5, e_t^6, e_t^7)$:

- $e_t^1 \in [0, 1]$, refers to energy level at time t.
- $e_t^2 \in [0, 70]$[1], refers to temperature at time t.
- $e_t^3 \in \{0, 1\}$, refers to whether the robot is in vertical position or not at time t.
- $e_t^4 \in [0, 100]$[2], refers to the presence of noise at time t.
- $e_t^5 \in \{0, 1\}$, refers to whether the robot detects or not an identified user within a scene at time t.
- $e_t^6 \in \{0, 1\}$, refers to whether the robot is touched or not at time t.

[1]Operating temperature (in °C) range according to the provided technical datasheet.

[2]In dB. Source: "Decibel Table - SPL - Loudness Comparison Chart". Retrieved 5 Mar 2012.

- $e_t^7 \in [0, 2000]$[3], refers to the light level captured at time t.

We assume conditional independence for the seven environmental variables, so that

$$p(e_t \mid b_t, e_{t-1}, e_{t-2}) = \prod_{i=1}^{7} p(e_t^i \mid b_t, e_{t-1}^i, e_{t-2}^i).$$

We describe now the evolution models for each of the environmental variables.

Energy level model We shall assume that $p(e_t^1 \mid b_t, e_{t-1}^1, e_{t-2}^1) = p(e_t^1 \mid b_t, e_{t-1}^1)$. We just need to know the current energy level and the action of the user (whether she plugged in or not the robot) to forecast the energy level. Indeed, we shall assume that

- If b_t = *discharge*, $e_t^1 = e_{t-1}^1 - k_1 \Delta t$, where k_1 is the energy consumption rate.
- If b_t = *recharge*, $e_t^1 = e_{t-1}^1 + k_2 \Delta t$, where k_2 is the energy recharging rate.

Δt represents the time difference between instants t and $t+1$. Based on trial and error, variables k_1 and k_2 have been fixed to 0.02 and 0.04 respectively, as it requires more time to discharge the battery than to charge it.

Temperature model We shall assume that $p(e_t^2 \mid b_t, e_{t-1}^2, e_{t-2}^2) = p(e_t^2 \mid e_{t-1}^2, e_{t-2}^2)$, as we are not able to detect the user's actions concerning temperature changes. We shall assume a simple model, such as $e_t^2 = e_{t-1}^2 + (e_{t-1}^2 - e_{t-2}^2)\Delta t$. More sophisticated models would include error terms, but the previous one is sufficient for our purposes, taking into account our computational limitations.

Inclination model We shall assume the generic model $p(e_t^3 \mid b_t, e_{t-1}^3)$, being b_t = *attack* or *move* , the relevant user actions, as they imply at least a variation over the inclination sensor, see Section 2.4. This sensor detects only whether (1) or not (0) the robot is in vertical position. Then, we use the evolution matrix shown in Table 2.1, where, depending on whether the robot was in vertical position or not

[3]In lux. According to the CIBSE (Chartered Institute of Building Services Engineers) Code for Lighting Part (2002).

and whether the robot inferred that the user action was *attack*, *move* or another, it will predict the next value of the inclination sensor (e_t^3).

e_{t-1}^3	Attack or Move at t	Neither Attack nor Move at t
0	0	0
1	0	1

Table 2.1: Evolution of being in vertical position.

For example, suppose that the environmental variable e_{t-1}^3 is 0, which means that the robot was not in vertical position at $t-1$, possibly because there was an *attack* or it was *moved* at time $t-1$. Let us now suppose that, at time t, the robot infers an *stroke* action, so that, according to Table 2.1, e_t^3 will still be 0, assuming that there has not been any change in the inclination level.

Presence of noise We assume the generic model $p(e_t^4 \mid b_t)$, with $b_t = shout$, the relevant user action. The noise sensor detects the level of noise surrounding the robot. Then, we use the evolution matrix shown in Table 2.2, where, depending on whether the robot inferred the user action *shout* or another, it will predict whether or not the next value of the noise sensor (e_t^4) is above the comfort zone,

$b_t = shout$	$b_t \neq shout$
1	0

Table 2.2: Evolution of being in a noisy environment.

in case it is, $e_t^4 = uth_{noise}$ +1, where uth_{noise} is the noise upper threshold defined as 50 dB [4] impacting negatively on the preference model; in case it is not above the comfort zone, $e_t^4 = e_{t-1}^4$.

As an example, suppose the robot inferred that it was *shouted* at time t, so that the environmental variable e_t^4 would be $uth_{noise} + 1$, which would impact on the robot's preference model.

[4]Source: "Decibel Table - SPL - Loudness Comparison Chart". Retrieved 5 Mar 2012.

Presence of an identified user We asume the generic dependence model $p(e_t^5 \mid b_t, e_{t-1}^5)$, with $b_t \in$ *interacting actions subgroup*, as those are the actions which require the presence of the user. The detection system probabilistically determines the presence of an identified user[5]. The output of this system shows the probability of a given user to be identified. If it is above 50 %, the user is considered as present within the scene. Adopting the evolution matrix shown in Table 2.3, we will predict whether, or not, there will be an identified user in the scene at the next instant. It will depend on whether the robot detected a user or not and whether the robot inferred a user action belonging to the subgroup of interacting actions or another. In this case, p_1 is the probability of detecting the user presence when b_t is not in the interacting actions subgroup. It will follow a $Beta - Binomial$ model, see Ríos Insua *et al.* (2012), with a $Beta(1,1)$ prior,

e_{t-1}^5	$b_t \in$ *interacting actions subgroup*	$b_t \notin$ *interacting actions subgroup*
0	1	0
1	1	p_1

Table 2.3: Evolution of the detection system.

$$p_1 | D_t \sim Beta(1 + x_1, 1 + n_1 - x_1),$$

with n_1 the number of ocurrences, and x_1 those in which the user has been detected. If necessary, it may be summarized through

$$\hat{p}_1 = E(p_1 \mid D_t) = \frac{1 + x_1}{2 + n_1}.$$

As an example, imagine a situation in which the robot has recognized the presence of the user at time $t-1$ ($e_{t-1}^5 = 1$) and inferred a *recharge* action. Then, according to Table 2.3, e_t^5 would be estimated through the probability p_1, as above.

[5] Adversary identification is not a core element of our work. We have based user identification on an eigenface recognition algorithm, implemented with OpenCv libraries, see Hewitt (2007).

Variations on the touch sensor We assume the generic model $p(e_t^6 \mid b_t, e_{t-1}^6)$, being $b_t = stroke$ or $b_t = move$, the relevant user actions. The touch sensors detect whether (1), or not (0), the robot has been touched. We use the evolution matrix shown in Table 2.4, where, depending on whether the robot was touched or not at $t-1$, and whether the robot inferred the user actions *stroke* or *move* or another, it will predict the next value of the touch sensor (e_t^6).

e_{t-1}^6	$b_t = stroke$ OR $b_t = move$	$b_t \neq stroke$ AND $b_t \neq move$
0	1	0
1	1	p_2

Table 2.4: Evolution of being touched.

Again, p_2, the probability of being touched when the action is neither *stroke*, nor *move*, follows a $Beta - Binomial$ model, with a $Beta(1,1)$ prior, and

$$p_2 | D_t \sim Beta(1 + x_2, 1 + n_2 - x_2),$$

being n_2 the number of ocurrences, and x_2 those in which the user has not touched the robot. It may be summarized through

$$\hat{p}_2 = E(p_2 \mid D_t) = \frac{1 + x_2}{2 + n_2}.$$

As an example, imagine an scenario in which the robot has detected no touch at time $t-1$ ($e_{t-1}^6 = 0$) and inferred a *move* action. According to Table 2.4, the environmental variable e_t^6 would be 1 at time t.

Variations on the light sensor We assume the generic model $p(e_t^7 \mid b_t)$, being $b_t = blind$, the relevant user action. The light sensor detects the discretized intensity of light surrounding the environment. We use the evolution matrix shown in Table 2.5, where, depending on whether the robot inferred the user action *blind* or another, it will predict the next value of the light sensor.

$b_t = blind$	$b_t \neq blind$
0	e^7_{t-1}

Table 2.5: Evolution of being blind.

As an example, suppose an scenario in which the robot has inferred a *flatter* action. Then, according to Table 2.5, the forecasted intensity of the light at time t (e^7_t) would remain as it was at time $t-1$.

2.4.3 Multiobjective preference model

We introduce now the preference model. As described in Section 2.3.2, the robot aims at satisfying five objectives, see Fig. 2.4, which, as in Maslow (1943), are ordered hierarchically by importance. They are:

- A primary objective concerning being properly charged.
- A secondary objective concerning being secure.
- A third objective concerning being taken into account by the user.
- A fourth objective concerning being accepted by the user.
- A fifth objective referring to being updated.

This hierarchy entails that the robot will invest most resources in achieving a sufficient level in the lowest objective, because of its higher weight. Once it has attained a sufficient value, it will redistribute its resources to achieve the next level, and so on. We describe now the global utility function used and the corresponding component utility functions.

Utility function

The pyramid's objectives is formalised through the hierarchy in Fig. 2.5, which includes the corresponding attributes from which we may deduce the sensors used to asses them, see Table 2.6. For example, the second objective, concerning security, takes into account whether the noise, temperature and light levels are

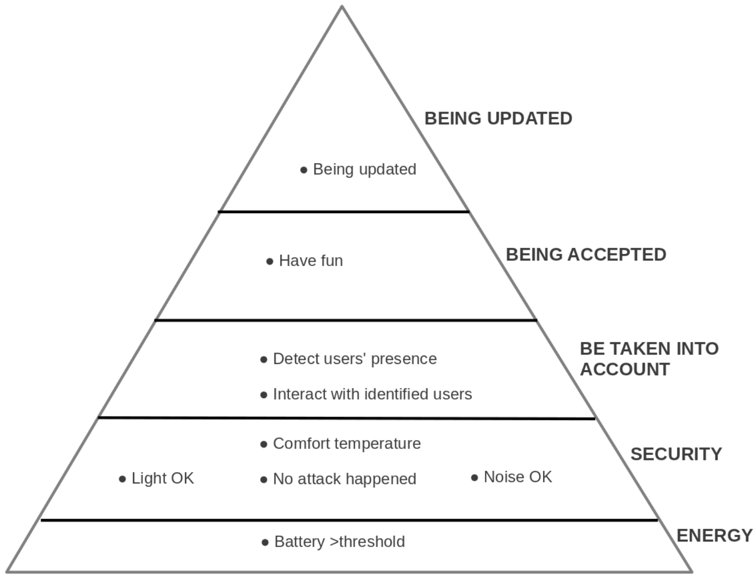


Figure 2.4: Pyramid of objectives.

appropriate and whether the robot is not being attacked. Therefore, the relevant sensors are the microphone to capture the noise level and a verbal attack (b_5 = offense); the temperature sensor, the light sensor and the inclination sensor, to verify whether the robot is being attacked or not, as described in Section 2.4.1.

Based on these five objectives, the global utility function will be

$$w_1 \times u_1(energy) + w_2 \times u_2(security) +$$

$$+ w_3 \times u_3(be\ taken\ into\ account) + w_4 \times u_4(being\ accepted) + w_5 \times u_5(being\ updated),$$

with $w_1 >> w_2 >> w_3 >> w_4 >> w_5 > 0$ and $w_1 + w_2 + w_3 + w_4 + w_5 = 1$, to stress the hierarchical nature of the objectives. We have initialized the weights as $w_1 = 0.3,\ w_2 = 0.25,\ w_3 = 0.2,\ w_4 = 0.15,$ and $w_5 = 0.1$.

Component utility functions

Objective 1: Energy The most important objective pays attention only to the energy level, e^1, measured in a scale [0,1]. The robot aims at having a sufficient energy level to perform its activities. A very low energy level is perceived as harmful by the robot. A sufficiently high energy level is good for the robot. We

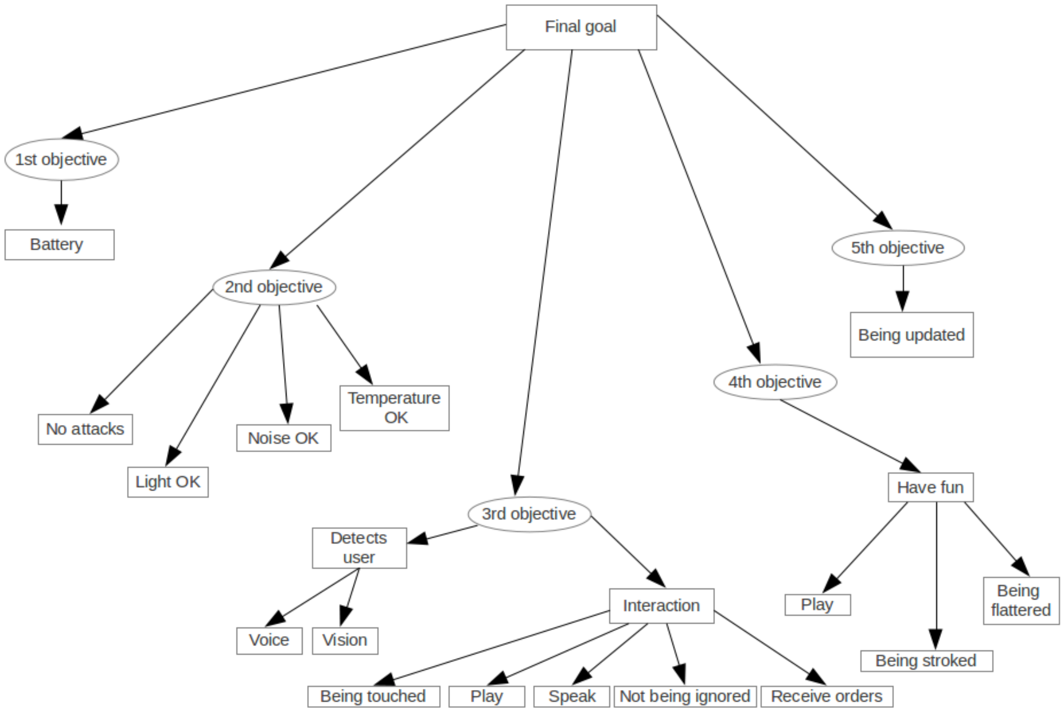


Figure 2.5: Objectives hierarchy, with first and second level objective and attributes.

represent it through

$$u_1(energy) = \begin{cases} 0, \text{ if } e^1 \leq lth_{energy} \\ 1, \text{ if } e^1 \geq uth_{energy} \\ \frac{e^1 - lth_{energy}}{uth_{energy} - lth_{energy}}, \text{ otherwise.} \end{cases}$$

We shall fix the upper threshold as $uth_{energy} = 0.5$ and the lower threshold as $lth_{energy} = 0.1$: we consider being above the 50 % of charge being still positive, and being below 10 % a risk situation.

Objective 2: Security The second objective refers to security. It essentially takes into account whether the robot is being attacked (either $b_5 = attack$ or $b_6 = offend$) by the user and whether it is functioning at an appropiate temperature. Secondarily, it pays attention to having appropriate light and noise levels

Objective	Name	Sensors
1.1	Energy	Battery level + wire plugin
2.1	No Attacks	Inclination + Microphone(ASR)
2.2	Light OK	Camera (Light reading)
2.3	Noise OK	Microphone (Noise reading)
2.4	Temperature OK	Thermometer
3.1.1	User Detection (Voice)	Microphone (ASR)
3.1.2	User Detection (Vision)	Camera (OpenCv)
3.2.1	Interaction (Being Touched)	Touch sensor
3.2.2	Interaction (Play)	Microphone (ASR)
3.2.3	Interaction (Speak)	Microphone (ASR)
3.2.4	Interaction (Not Being Ignored)	Microphone (ASR)
3.2.5	Interaction (Receive Orders)	Microphone (ASR)
4.1	Have Fun (Play)	Microphone (ASR)
4.2	Have Fun (Stroked)	Microphone (ASR)
4.3	Have Fun (Flattered)	Microphone (ASR)
4.3	Have Fun (Apologized)	Microphone (ASR)
5.1	Being Updated	Database reading

Table 2.6: Sensors involved in the objectives hierarchy.

in the room. It is represented through

$$u_2(security) =$$

$$= w_{21} \times u_{21}(attack) + w_{22} \times u_{22}(temperature) + w_{23} \times u_{23}(light) + w_{24} \times u_{24}(noise),$$

with $\sum_{i=1}^{4} w_{2i} = 1$, and weights ordered in importance as follows: $w_{21} > w_{22} >> w_{23} > w_{24} > 0$. Weights are initialized as $w_{21} = 0.5$, $w_{22} = 0.25$, $w_{23} = 0.15$ and $w_{24} = 0.1$.

The component utility functions are

$$u_{21}(attack) = \begin{cases} 1, & \text{if no attack inferred at } t, \text{ neither at } t-1 \\ 0.5, & \text{if after an attack at } t-1, \text{ there was no attack at } t \\ 0, & \text{otherwise,} \end{cases}$$

where attack is referred to actions b_5=*attack* and b_6=*offend*.

$$u_{22}(temperature) = \begin{cases} 0, \text{ if } e^2 < lth_{temp} \text{ or } \ e^2 > uth_{temp} \\ 1, \text{ if } \ \ lcth_{temp} < e^2 < ucth_{temp} \\ \\ 1 - \left(\frac{lcth_{temp} - e^2}{lcth_{temp}}\right), \ \text{ if } \ e^2 \leq lcth_{temp} \\ \\ \frac{uth_{temp} - e^2}{uth_{temp} - ucth_{temp}}, \ \text{ if } e^2 \geq ucth_{temp}, \end{cases}$$

with $lth_{temp} = 0°$ C, $uth_{temp} = 35°$ C, $lcth_{temp}$ (lower thermal comfort) $= 20°$ C and $ucth_{temp}$ (upper thermal comfort) $= 25°$ C, where e^2 is the temperature.

As mentioned in Section 2.4.2, the light sensor detects the discretized intensity (e^7) of light in the environment. We use

$$u_{23}(light) = \begin{cases} 0, \text{ if } \ e^7 > uth_{light} \\ 1, \text{ if } \ \ lcth_{light} < e^7 < ucth_{light} \\ \\ 1 - \left(\frac{lcth_{light} - e^7}{lcth_{light}}\right), \ \text{ if } \ e^7 \leq lcth_{light} \\ \\ \frac{uth_{light} - e^7}{uth_{light} - ucth_{light}}, \ \text{ if } e^7 \geq ucth_{light}, \end{cases}$$

with $uth_{light} = 1500$ lux, $lcth_{light}$ (lower lighting comfort) $= 200$ lux and $ucth_{light}$ (upper lighting comfort) $= 500$ lux.

Finally, the noise sensor detects the intensity (e^4) of the noise surrounding the environment. We use

$$u_{24}(noise) = \begin{cases} 0, \text{ if } e^4 \geq uth_{noise} \\ 1, \text{ if } \ e^4 \leq lth_{noise} \\ \\ 1 - \left(\frac{e^4 - lth_{noise}}{uth_{noise} - lth_{noise}}\right), \ \text{ otherwise}, \end{cases}$$

with $lth_{noise} = 10$ dB [6] and $uth_{noise} = 50$ dB.

[6]Source: "Decibel Table - SPL - Loudness Comparison Chart". Retrieved 5 Mar 2012.

Objective 3: Being taken into account The third objective is related with being taken into account by the user. It evaluates whether the owner is around it and whether she is interacting with it by being touched or asking the robot to play, ordering something, starting a conversation or, simply, not ignoring it. We represent it through the component utility function

$$u_3(\textit{be taken into account}) = w_{31} \times u_{31}(\textit{interaction}) + w_{32} \times u_{32}(\textit{detection}),$$

with $\sum_{i=1}^{2} w_{3i} = 1$, and weights ordered in importance as follows: $w_{31} >> w_{32} > 0$. Weights may be initialized as $w_{31} = 0.7$ and $w_{32} = 0.3$. We further decompose u_{31} according to:

$$u_{31}(\textit{interaction}) = w_{311} \times u_{311}(\textit{not ignored}) + w_{312} \times u_{312}(\textit{being spoken}) +$$

$$+ w_{313} \times u_{313}(\textit{asked to play}) + w_{314} \times u_{314}(\textit{being ordered}) + w_{315} \times u_{315}(\textit{being touched}),$$

with $\sum_{i=1}^{5} w_{31i} = 1$, and weights ordered in importance as follows: $w_{311} > w_{312} > w_{313} > w_{314} > w_{315} > 0$. Those weights may be initialized as $w_{311} = 0.3$, $w_{312} = 0.25$, $w_{313} = 0.2$, $w_{314} = 0.15$ and $w_{315} = 0.1$.

The corresponding component utility functions are:

$$u_{311}(\textit{not ignored}) = \begin{cases} 0, & \text{if the agent is ignored at } t \\ 0.5, & \text{if it was ignored at } t-1, \text{ but was not at } t \\ 1, & \text{otherwise,} \end{cases}$$

$$u_{312}(\textit{being spoken}) = \begin{cases} 1, & \text{if a grammar has been started at } t \\ 0.5, & \text{if a grammar was started at } t-1 \\ 0, & \text{otherwise,} \end{cases}$$

where *being spoken* meaning that the user has started dialogue containing words within a specific set [standard-issues] in order to interact with the agent, e.g. how its day was. Such sets are identified by an interpreter, as ChatScript, see Wilcox (2013), and consists on a set of rules that let the robot follow the guidelines of the user in a speaking context, see Section 1.1. For instance, if the user says *tell* *me about your day*, the robot will identify the verb: *to tell*, and the issue: *your day*, and will look for a response that properly matches both of them.

$$u_{313}(asked\,to\,play) = \begin{cases} 1, & \text{if the robot is asked to play by the user} \\ 0.5, & \text{if the robot was asked to play by the user at } t-1 \\ 0, & \text{otherwise,} \end{cases}$$

where *asked to play* refers to detecting an order to play from the user, including the game's title.

$$u_{314}(being\ ordered) = \begin{cases} 1, & \text{if the robot receives an order} \\ 0.5, & \text{if the robot received an order at } t-1 \\ 0, & \text{otherwise,} \end{cases}$$

where *being ordered* consists of detecting an order among a catalogue of actions within a certain grammar. *Being ordered, asked to play* or *being spoken* are evaluated through the ASR algorithm, see Section 1.1, so they depend on the defined grammar and are detected in a probabilistic way.

$$u_{315}(being\ touched) = \begin{cases} 1, & \text{if } e^6 \ > \ uth_{touched} \\ 0, & \text{otherwise,} \end{cases}$$

with $uth_{touched} = 50\%$ of probability of being touched, see Table 2.4.

With respect to u_{32}, we shall use

$$u_{32}(detection) = \begin{cases} 1, & \text{if } e^5 \ > \ uth_{detected} \\ 0, & \text{otherwise,} \end{cases}$$

with $uth_{detected} = 60\ \%$ of probability of detecting a user, see Table 2.3.

Objective 4: Being accepted The fourth objective is aimed at evaluating whether the robot is being accepted by the user, whether she is interacting and having fun with it. We represent this through

$$u_4(\textit{being accepted}) = w_{41} \times u_{41}(\textit{play}) + w_{42} \times u_{42}(\textit{flatter}) +$$

$$+ w_{43} \times u_{43}(\textit{stroke}) + w_{44} \times u_{44}(\textit{apologize}),$$

with $\sum_{i=1}^{4} w_{4i} = 1$, and weights ordered in importance as follows: $w_{41} > w_{42} > w_{43} > w_{44} > 0$. Weights may be initialized as $w_{41} = 0.35$, $w_{42} = 0.3$, $w_{43} = 0.2$ and $w_{44} = 0.15$. The component utility functions are:

$$u_{41}(\textit{play}) = \begin{cases} 1, & \text{if the user wants to play at } t \\ 0.5, & \text{if she wanted to play at } t-1, \text{ but not at } t \\ 0, & \text{otherwise,} \end{cases}$$

where we are considering that the user wants to play whenever she performs b_{14}=*play*.

$$u_{42}(\textit{flatter}) = \begin{cases} 1, & \text{if the user flatters the agent at } t \\ 0.5, & \text{if she flattered it at } t-1, \text{ but not at } t \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{43}(\textit{stroke}) = \begin{cases} 1, & \text{if the user strokes the agent at } t \\ 0.5, & \text{if she stroked it at } t-1, \text{ but not at } t \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{44}(\textit{apologize}) = \begin{cases} 1, & \text{if the user apologizes to the agent at } t \\ 0.5, & \text{if she apologized at } t-1, \text{ but not at } t \\ 0, & \text{otherwise,} \end{cases}$$

Objective 5: Being updated Finally, at the fifth level objective, the robot checks whether it has been updated recently. Our current implementation of such component utility function is

$$u_5(\textit{being updated}) = \begin{cases} 1, & \text{if robot version date } < \text{ 2 months ago} \\ 0, & \text{otherwise.} \end{cases}$$

Each time the robot is rebooted, it will check the date of its current version to evaluate whether it is updated or not.

2.5 Assessing the model

We have developed a simulator in Python to study the behavior of the model. Two sets of experiments were performed. The first one was aimed at assessing the forecasting performance of our model; the second one was aimed at assessing the decision-making aspects.

2.5.1 First experiment

As mentioned, our first experiment tries to evaluate how model averaging techniques work, observing how $p(M_1 \mid D_t)$ (and $p(M_2 \mid D_t)$), as defined in Section 2.4.2, evolve depending on how reactive the robot considers the user to be to its actions. In other words, we aim at testing whether our model uncovers the type of user the robot is facing. To do so, within each iteration, see loop in Fig. 2.1, we have simulated the user action b_t for the user model given b_{t-1} and b_{t-2}, and the user action b_t for the classical conditioning model given a simulated a_t. The information is stored, within the corresponding data structures.

We have designed three different scenarios for this purpose:

- In the first scenario, we simulate responses consistent with model M_1, i.e. the user performs to be independently of the agent's actions, and try to assess whether our forecasting model uncovers M_1.

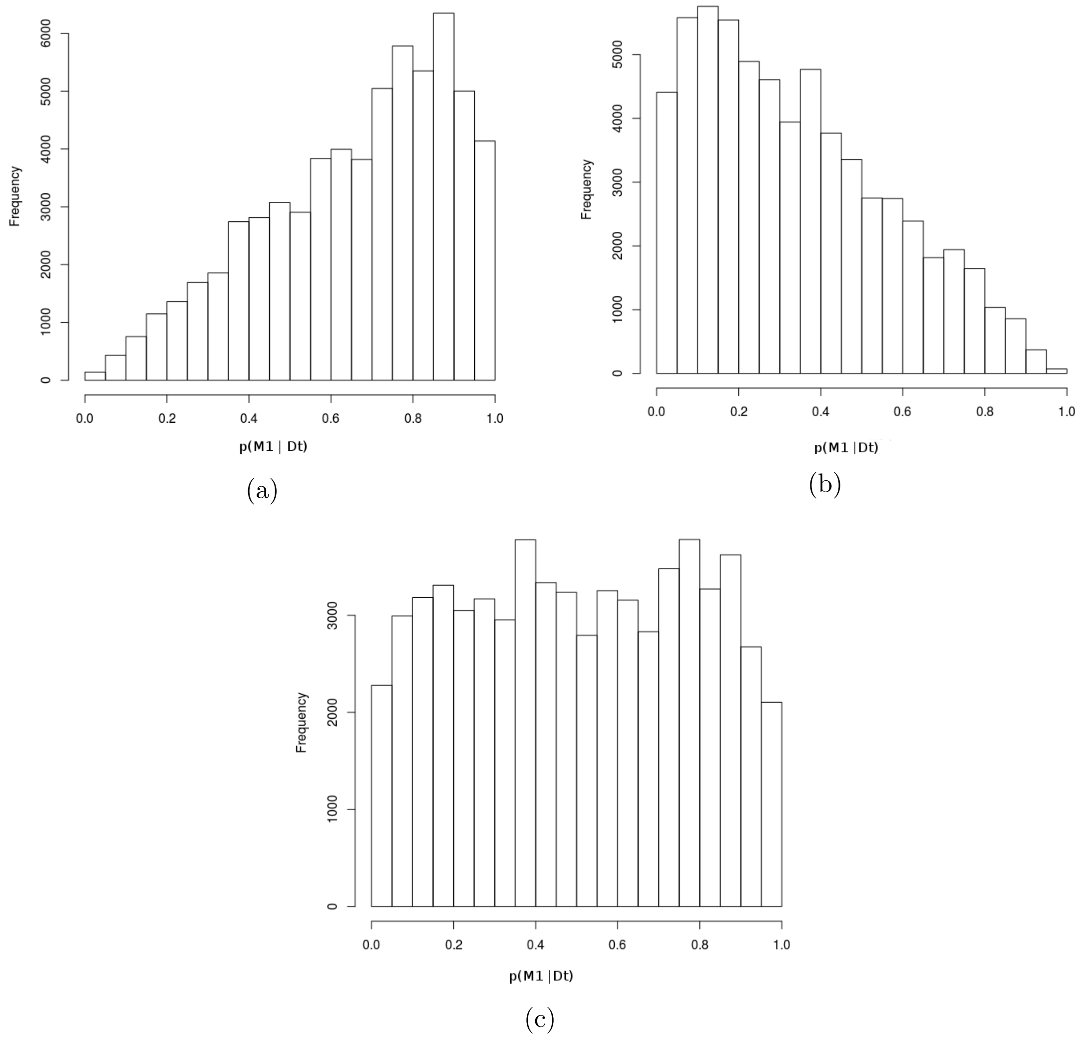
- In a second one, we simulate responses consistent with model M_2, i.e. the user performs her actions depending on the agent's actions, trying to assess whether M_2 is uncovered by our forecasting model.

- For the third scenario, we simulate responses combining both models, with 0.5 probability for each model, in order to evaluate the robustness of our forecasting model.

	First Scenario M_1	Second Scenario M_1	Third Scenario M_1
min.	0.0051	0.0038	0.0034
1^{st} quartile	0.4946	0.1481	0.2621
median	0.7064	0.3040	0.4982
3^{rd} quartile	0.8492	0.5007	0.7485
max.	0.9990	0.9920	0.9989

Table 2.7: Summary of the first experiment data.

Table 2.7 provides a summary of the results for the three scenarios, providing the details of the obtained data. We can see the histograms for $p(M_1|D_t)$, over 60000 iterations in Fig. 2.6. As we may see under the first scenario, see Fig. 2.6(a), we tend to identify M_1 as more likely, as expected, as our agent estimates the user to perform her actions following her own behavior independently of our agent's actions. Similarly, under the second scenario, see Fig. 2.6(b), we tend to identify M_2 as the most likely. In this case, our agent estimates that the user is performing being reactive to those actions implemented by our agent. Finally, under the third scenario, see Fig. 2.6(c), we get an equilibrium between M_1 and M_2, which means that our agent considers that the user is behaving sometimes reactive and others independently of its actions.



(a) (b) (c)

Figure 2.6: Experiment to prove model averaging techniques.

2.5.2 Second experiment

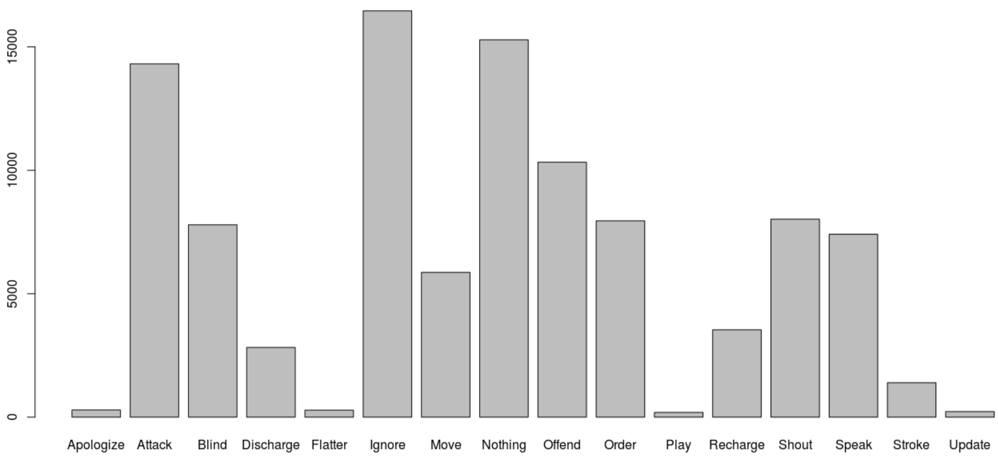
In our second experiment, we assess the decision-making performance of our model. We actually study the impact of weights on the agent decision-making. To do so, we need to simulate the environment in which the robot is placed and different user behaviors. Then, we compare the impact of different objectives' weights on the agent decision-making.

To simulate the user's behavior, we make the following assumptions

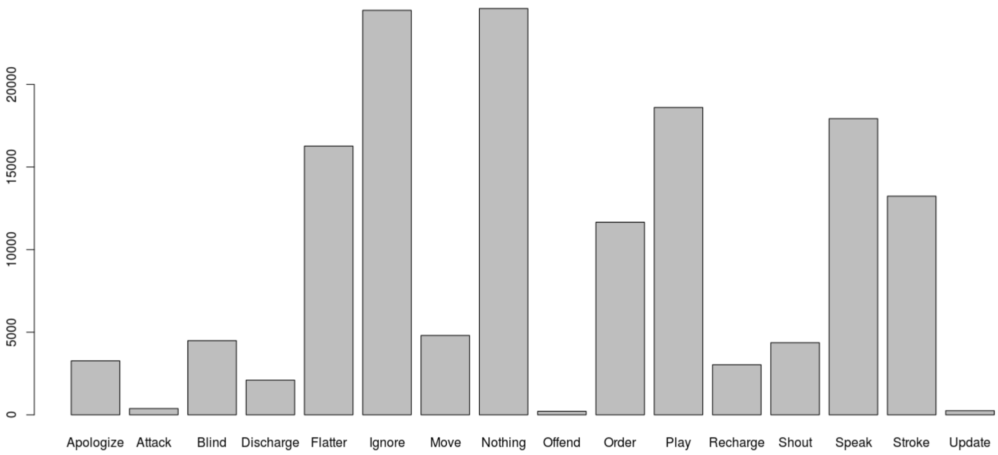
- Whenever the agent performs a_3 = *ask for charging*, the user will perform b_1 = *recharge*.
- At least, the user will perform b_9 = *update* once per 10.000 loop iterations.
- 50% of the times, the user will behave reactively to our agent's actions. In that case, she will act consequently with the agent, following two different behaviors: benevolent and malevolent, see Fig. 2.7. As an example, if the agent performs a_4 = *complain*, in the first case, the user would randomly choose an action among the set {*flatter, speak, play, stroke, apologize, recharge, ignore* and *do nothing*}, looking for cheering the agent up. In the second case, under the same agent action the user would consider the set {*move, attack, ignore, offend, blind, shout, order* and *do nothing*} punishing the agent.
- Otherwise, the action performed by the user will be randomly generated from a Multinomial-Dirichlet distribution.

As explained in Section 2.3.3, agent's actions are chosen in a randomized way, with probabilities proportional to the estimated predictive expected utilities. However, we introduce certain rules that must be satisfied before that randomization:

- Action a_1: *ask for help* will be considered only if the robot feels insecure, i.e., the component utility function of the second objective is under 0.5, see Section 2.4.3.
- Action a_2: *salute* will be taken into account only if the robot detects a new user in the scene.



(a)



(b)

Figure 2.7: Distribution of actions of different user behaviors: malevolent (a) and benevolent (b) users.

- Action a_3: *ask for charging* will be taken into account only if the battery level is under 20 %.
- Action a_4: *complain* will be considered only if the utility of the consequences is under 0.4.
- Action a_5: *play* will be considered only if the user has taken into account action b_{11} = *ask for playing*.
- Action a_8: *ask for shutting down* will be considered only if the utility of the last consequences faced is under 0.15.
- Action a_{11}: *obey* will be considered only if the user has taken into account action b_{12}= *order*.

If the conditions corresponding to any of these actions are not satisfied, the robot will not include them within the set of available actions for randomization.

The initial conditions of the environment have been set up to

- Battery (e^1): 75 %.
- Temperature (e^2): 25° C.
- Inclination (e^3): Vertical position.
- Noise in the environment (e^4): 50 dB.
- Presence of an identified user (e^5): No.
- Detection on any touch sensor (e^6): No touch.
- Light intensity in environment (e^7): 500 lux.

They will evolve as described in Section 2.4.2.

Initial conditions The initial weights tested are: $w_1 = 0.3$, $w_2 = 0.25$, $w_3 = 0.2$, $w_4 = 0.15$ and $w_5 = 0.1$. This represents a state in which our robotic agent shall give a balanced importance to its objectives. They are ordered hierarchically, but with no large differences among them.

Figs. 2.8 and 2.9 represent the number of times each agent action (rows) was performed in response to the observed user action (columns). We may appreciate that in the case of a malevolent user, Fig. 2.8, the agent performs more frequently actions like *complain* (a_4) or *ask for charging* (a_3) in comparison to when it faces a benevolent user, see Fig. 2.9, in which, the *apologize* action (a_{14}) increases due to the user's behavior.

Figure 2.8: Agent's actions depending on the user's action regarding a malevolent user.

Fig. 2.10 shows the utilities obtained from the consequences. As we may appreciate, the utilities obtained facing a benevolent user, Fig. 2.10(b), are higher (first quartile ≈ 0.446, median ≈ 0.507, third quartile ≈ 0.593) than when facing a malevolent user (first quartile ≈ 0.328, median ≈ 0.382, third quartile ≈ 0.427), see Fig. 2.10(a).

Figure 2.9: Agent's actions depending on the user's action regarding a benevolent user.

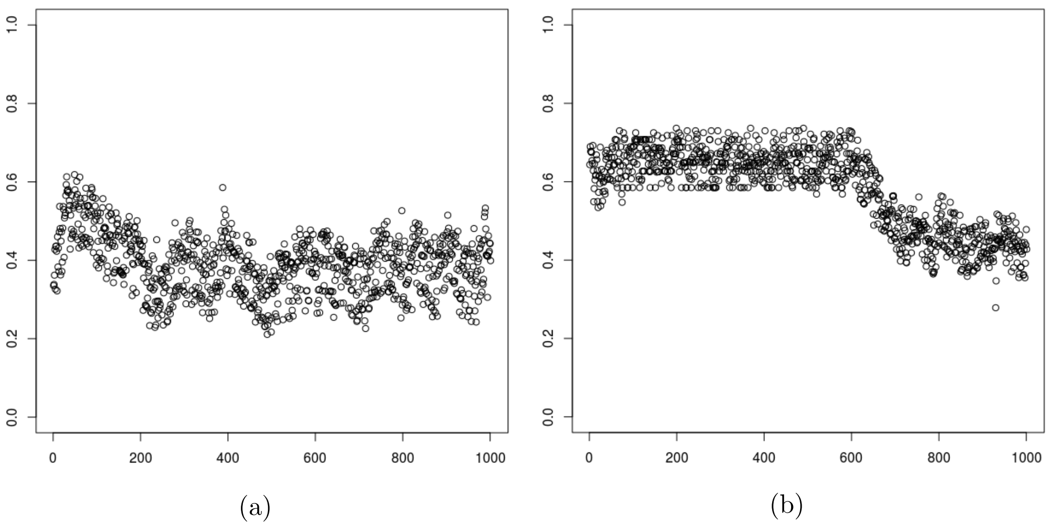
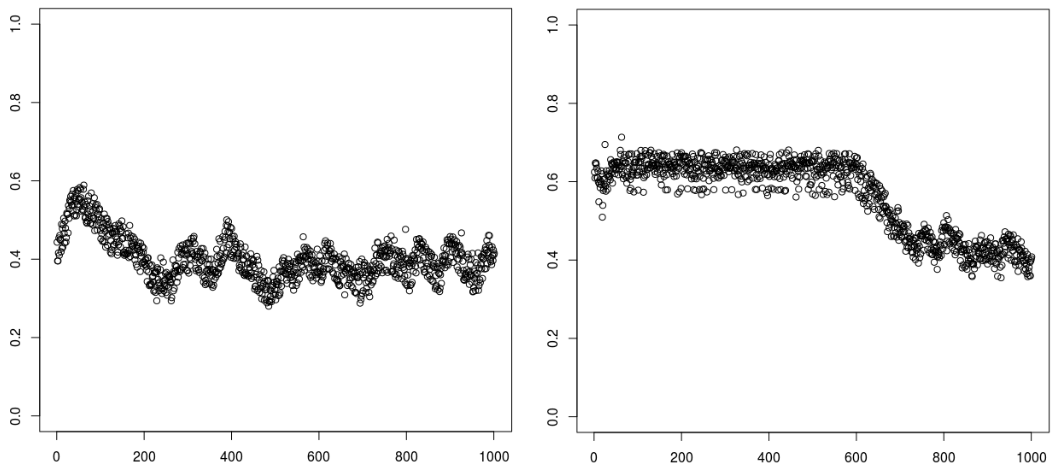


Figure 2.10: (a) refer to malevolent behavior, whereas (b) to benevolent.

The graphs in Fig. 2.11 shows the expected utilities of the agent's optimal alternative. As before, the expected utility computed facing a benevolent user is higher (first quartile ≈ 0.432, median ≈ 0.485, third quartile ≈ 0.576), see Fig. 2.11(b), than when facing a malevolent user (first quartile ≈ 0.355, median ≈ 0.383, third quartile ≈ 0.414), see Fig. 2.11(a).



(a) (b)

Figure 2.11: (a) refer to malevolent users, whereas (b) to benevolent ones.

First variation We increase the priority of the lower objectives, modifying the weigths to $w_1 = 0.7$, $w_2 = 0.2$, $w_3 = 0.09$, $w_4 = 0.007$ and $w_5 = 0.003$, representing a robotic agent which is really concerned about its primary needs.

Figs. 2.12 and 2.13 represent each agent action (rows) in response to the observed user action (columns). As before, we may notice that when our agent is dealing with a malevolent user, see Fig. 2.12, the agent performs more frequently actions like *complain* (a_4) and *ask for charging* (a_3). When dealing with a benevolent simulated user, see Fig. 2.13, the frequency of *apologize* action (a_{14}) increases, as well as that of *flatter* (a_{12}). Comparing the frequency of actions with respect to the variation of the weights, there is no action that is generally more or less performed than others. However, in those cases in which a benevolent user performs a specific action, like *attack*, the agent tends to implement a certain set of actions more often than others. In the mentioned case, actions like *ask for help*

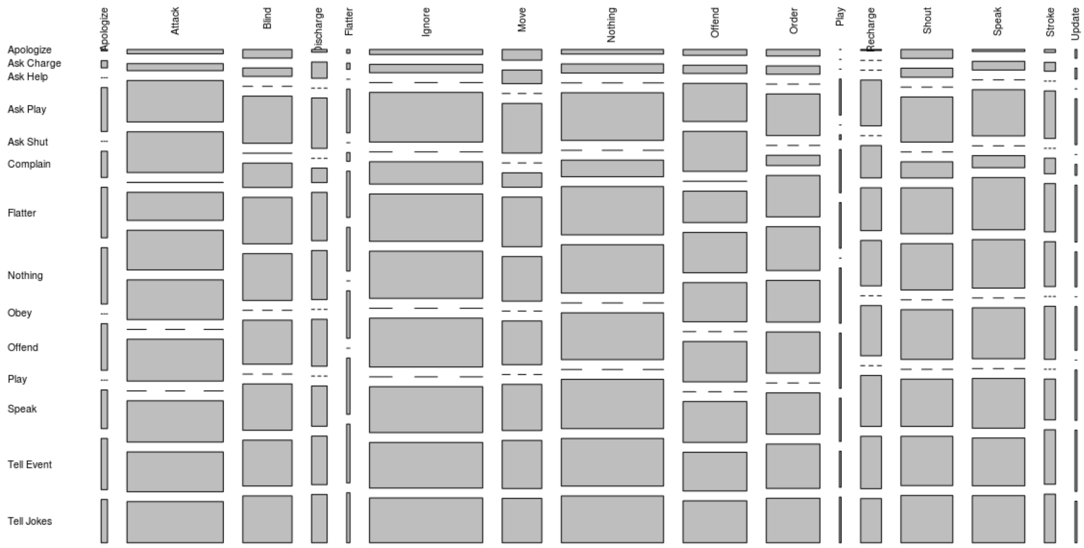


Figure 2.12: Agent's actions depending on a malevolent user's action.

(a_1) are more repeated than under the initial weights, while actions like *offend* (a_{13}) are less repeated, see Fig. 2.13. We may conclude that, under variations on model weights, agents modify their behavior.

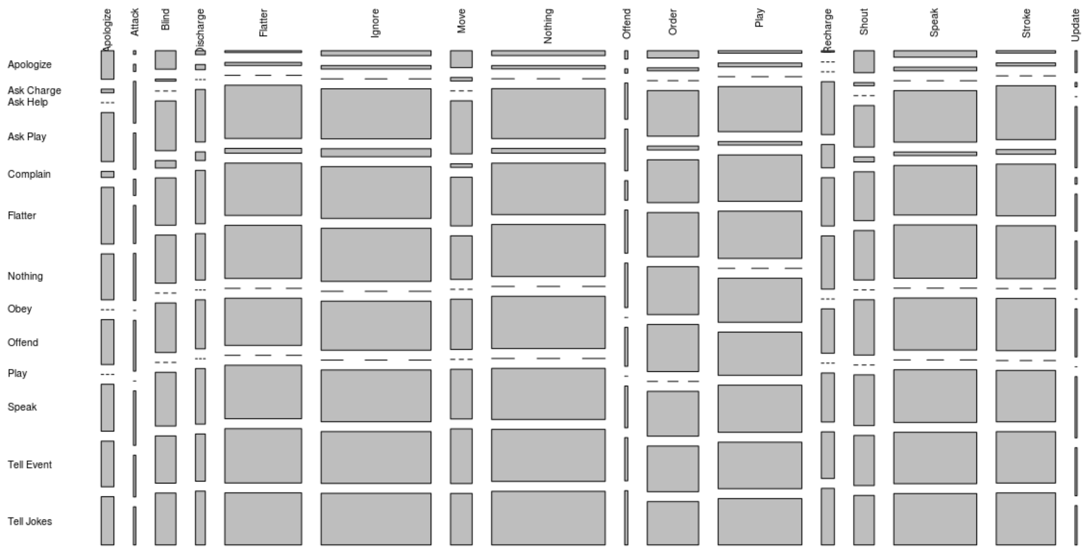
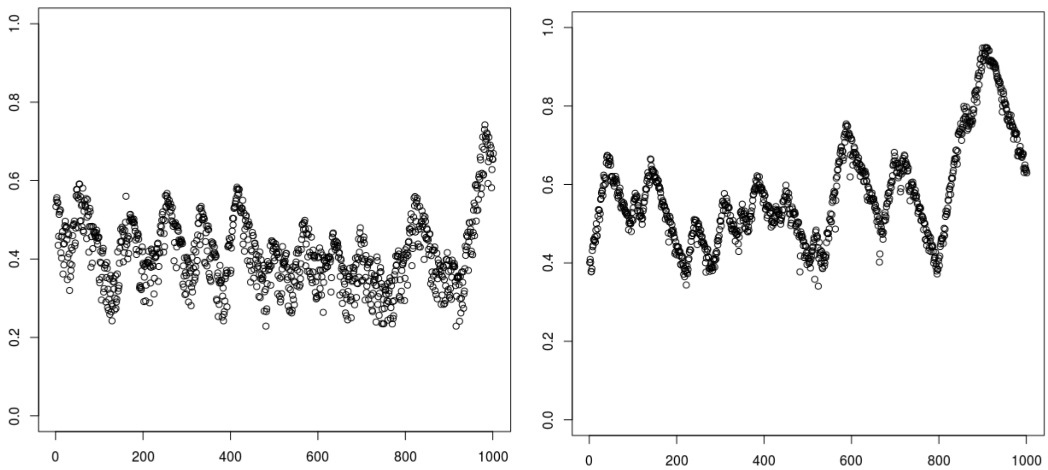


Figure 2.13: Agent's actions depending on a benevolent user's action.

Fig. 2.14 shows the utilities from the consequences. As before, we may notice that the utilities obtained facing a benevolent user, see Fig. 2.14(b), are higher (first quartile ≈ 0.439, median ≈ 0.54, third quartile ≈ 0.7) than when facing a malevolent user (first quartile ≈ 0.344, median ≈ 0.4, third quartile ≈ 0.467), see Fig. 2.14(a). Comparing these utilities to those with initial weights, see Fig. 2.10, we may observe that the agent is more influenced by the user's behavior under this variation of the objectives' weights than under the initial weights, being bigger the interquartile range (IQR) on both cases: $0.259 > 0.146$, for the case in which the agent faces a benevolent user, and $0.123 > 0.098$, otherwise.



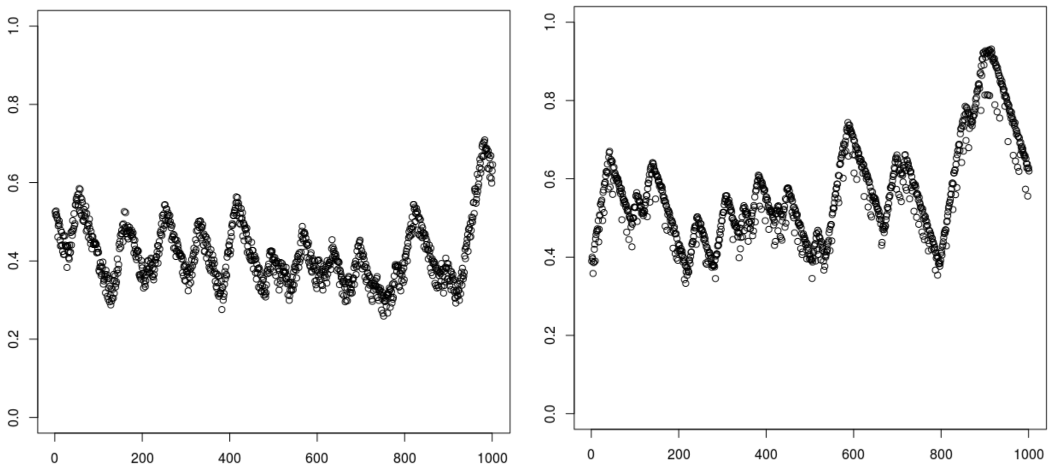
(a)

(b)

Figure 2.14: Utilities obtained from the consequences, facing a malevolent (a) and a benevolent user (b).

The graphs in Fig. 2.15 shows the expected utilities of the agent. Not surprisingly, the expected utility facing a benevolent user is higher (first quartile ≈ 0.428, median ≈ 0.527, third quartile ≈ 0.684), see Fig. 2.15(b), than when facing a malevolent user (first quartile ≈ 0.353, median ≈ 0.396, third quartile ≈ 0.464), see Fig. 2.15(a).

Making a comparison with the expected utilities attained with initial weights, see Fig. 2.11, we may observe that, as before, the agent is less influenced by the user's behavior under the initial weights than under this variation of the objectives weights, being smaller the interquartile range (IQR) on both cases: $0.256 > 0.144$, for the case in which the agent faces a benevolent user, and $0.11 > 0.059$, otherwise.



(a) (b)

Figure 2.15: Expected utilities obtained from the consequences, while facing a malevolent (a) and a benevolent user (b).

2.6 Implementation with AiSoy1

As explained above, we have implemented this model into an AiSoy1 robot which uses a Rapsberri Pi Model B as a main board. The Raspberri Pi has a Broadcom BCM2835 system on a chip including an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and 512 MB of RAM, see Raspberry Pi (2011). The board is connected to all the sensors and actuators.

To control this board, and those sensors and actuators from the AiSoy1, we have integrated ROS into it. ROS is a software framework capable of providing operating system-like functionalities such as hardware abstraction, message-passing between processes, and package management, see WillowGarage (2008).

To adapt ROS to our system, we have developed several ROS modules that allow us to communicate with the AiSoy1's hardware. The implementation design is based on a multilayered software architecture, in which:

- At the bottom lies the hardware layer composed of the Raspberri Pi board and the different sensors and actuators.
- Above that, there is the ROS layer in which we may find the ROS modules used to access the robot's hardware.
- Then, we find the SDK-API layer in charge of calling the ROS modules to access the hardware.
- Finally, the AiROS layer, which contains the described model, will call the SDK-API requesting information or sending orders.

Each layer can communicate exclusively with that adjacent to it. As an example, consider that our model is at the first step in the planning loop, see Fig. 2.1, so that it would like to read the incumbent information from the sensors. To do so, AiROS will call the SDK-API command *aisoy_sdk_sensor*, which will call the ROS node *aisoy_sensorimotor* to read from all the available sensors into the Raspberri Pi board.

When the robot is switched on, AiROS and the ROS node *aisoy_launch* are triggered. That ROS node will launch all the ROS nodes which will start listening to any message from the SDK-API layer. The main advantage of this architecture is the flexibility provided so that we do not have to modify any layer just in case we want to use a different AiROS module.

The functionality of the SDK-API and ROS nodes developed is the following:

- *aisoy_sdk_asr* and *aisoy_asr* are involved in the use of the ASR algorithm, capturing what the user is saying.
- *aisoy_sdk_camera* and *aisoy_camera* are used whenever we require to access the camera, to take a picture, identify a user, check the surrounding light level, etc.
- *aisoy_sdk_chatbot* and *aisoy_chatscript* complement the ASR component classifying words into grammar sets.
- *aisoy_sdk_common* and *aisoy_common* store all the programming variables initializing them.
- *aisoy_sdk_sensor* and *aisoy_sensorimotor* are involved in reading the sensors and obtaining information from them.
- *aisoy_sdk_tts* and *aisoy_tts* are the Text to Speech component which allow the robot talk.

2.7 Discussion

In this Chapter, we have described a level-1 ARA based behavioral model of an autonomous agent, which processes information from its sensors, facing a user based on multi-attribute decision analysis at its core, complemented by forecasting models of the adversary. They are place in an environment which changes under the user actions. Both, the agent and the user makes decisions within fixed finite sets of actions.

The agent is computationally limited which entails forecasting only one period ahead, with a limited memory. Through its forecasting models aims at predicting the evolution of the environment and the behavior of the user. Using its sensors, it will infer, deterministic or probabilistically, how the environment changes and the user acts. Through a preference model it evaluates the impact that such actions make on its goals, which are hierarchically ordered.

Through a set of experiments we have successfully demonstrated that: the developed forecasting models work as expected, and our agent modifies its behavior facing different users under different objectives' weights, acting in a natural way.

The model has been implemented within a RaspberryPi Model B, which is the same board included inside the AiSoy1. Two goals have been reached during this Chapter: designing a robust decision-making model to be implemented in non-expensive robotic platforms, and explaining how Adversarial Risk Analysis may be extended to a simple case in the robotic field.

In the next Chapter, we will see how the Adversarial Risk Analysis is extended to more complex scenarios, in which our agent faces several adversaries.

Chapter 3

Designing Societies of Robots

3.1 Introduction

In Chapter 2, we have described a behavioral model for an autonomous decision agent which processes information from its sensors, facing an adversary, using multi-attribute decision analysis at its core, complemented by models forecasting the decision-making of the adversary. We call this the basic Adversarial Risk Analysis framework. Both, agent and user, took actions from their corresponding finite sets of alternatives ($\mathcal{A}$ and $\mathcal{B}$). That model was implemented within an AiSoy1 robot, as presented in Section 2.6.

In this chapter, we refer to Multi-Agent Systems, see Wooldridge (2008), exploring the social needs of our robotic agent and how it handles interactions with several agents, both human and robotic ones. We have in mind different scenarios, see Fig. 3.1. In Fig. 3.1(a), we consider a single agent facing multiple adversaries, agents and users. In Figs. 3.1(b) and 3.1(c), several agents compete among themselves in their interaction with several users. Within interactive competitive scenarios, the agents face conflicts which deal with Adversarial Risk Analysis (ARA) models, which will be compared with Nash Equilibria concepts. Finally, there could be multiple agents cooperating to satisfy themselves and the users, see Fig. 3.1(d). In cooperative scenarios, a solution concept for cooperative games is introduced, in which agents will try to separate as much as possible from the competitive solution, while dominating it. Agents may move from a cooperative to a competitive attitude, and vice versa, depending on their previous interaction experience.

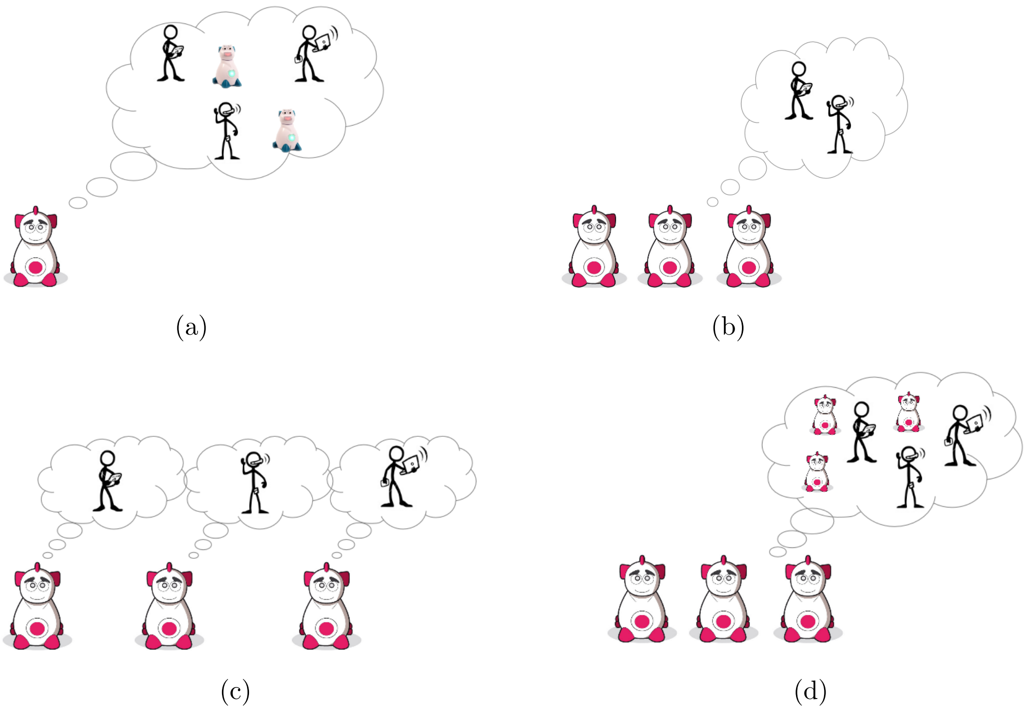


Figure 3.1: Different scenarios taken into account.

Throughout this chapter we explore the scenarios outlined. Our interest in modeling this problem is in the design of societies of autonomous robots which may cooperate or compete among themselves to support a group of persons, depending on the evolution of the environment they are in. As an example, consider three kids used to a collaborative working environment at school. At home, each of them is supported by his own edutainment robot in their weekly homework. The robots, which may communicate through Internet, replicate the school collaborative work environment, and would look for helping the children together. At some point, either because they receive an order from their users or because they perceive that they are performing inappropriately, they may decide to change their behavior turning into a competitive, cooperative or a mixed attitude. Those agents could be used within different environments as interactive robotic pets, robotic babysitters and teaching assistants or cooperative caregivers for the elderly.

The chapter is structured as follows. In Section 3.2, we consider a case in which a decision agent is identifying several users and robotic agents, and makes decisions depending on the adversary it is facing. Then, in Section 3.3, we define a case of a society of competitive robots which interact with humans, and a case in which several human-agent teams compete. For comparative purposes, we deal with them through both the game theoretic and the ARA frameworks.

In Section 3.4, we introduce a method to compute cooperative solutions within a society of cooperative agents. Then, we describe the evolution from a competitive to a cooperative attitude, see Section 3.5. We finally provide some computational experience in Section 3.6 and end up with some discussion.

3.2 Supporting an agent facing several agents and users

In this Section, we extend our basic model from Chapter 2 to a case in which the agent faces several adversaries, which may be agents or users, see Fig. 3.2. This model will be the starting point for later Sections. As an example, assume

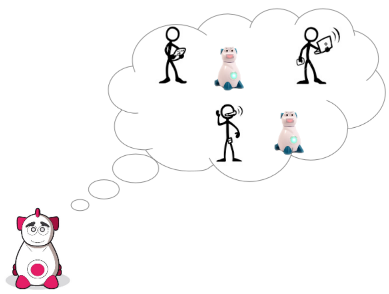


Figure 3.2: A single agent facing multiple adversaries (agents and users).

that our agent (A_1) is supporting two children (B_1 and B_2) in their daily school assignments. A_1 should be able to identify who is who to evaluate how correctly each of them is working, and deliver the corresponding score and support.

For that purpose, the agent must be able to identify the adversary it is facing and will have different forecasting models in relation with each of the known opponents. We assume that the agent will face just one adversary at each of the time steps of the scheme described in Fig. 3.3. Using some identification method,

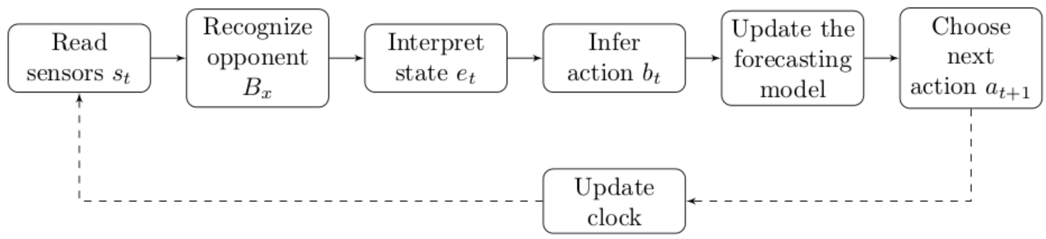


Figure 3.3: Agent loop with advesary recognition.

e.g. based on voice and/or vision, the agent will guess who is the user/agent it is dealing with and adapt its behavior accordingly. The difference between facing another agent or a user would essentially be the set of actions available for the corresponding adversary forecasting model. We do not consider adversary identification as a core element of our work. For that reason, we have based the identification of the opponent B_x on eigenface recognition algorithms, see Zhao *et al.* (2003) for a face recognition survey, and implemented it with OpenCv libraries, see Hewitt (2007). We assume that the agent computes the probabilities $p(B_x|D_t)$ of various possible adversaries B_x given the data D_t available, which for this problem will typically be an image from the adversary's face. Our agent will not identify physically other robotic agents as there is no physical difference among them, because we assume all robots are of the same type, but it will be possible to identify them, based on a communication protocol they maintain using their radio module.

3.2.1 Model

As in Chapter 2, our agent A_1 makes decisions within a finite set $\mathcal{A}$. In this case, there are q adversaries $B_1, \ldots, B_q$ which interact with A_1. An index x will be used to identify the corresponding adversary. B_x may be an agent or a user. She will make decisions within the set $\mathcal{A}$, in case she is an agent, or a set $\mathcal{B}$, in case she is a user.

The agent decision model is similar to that in Chapter 2. However in this case, the forecasting model is conditional on the guessed adversary, so that equation (2.1) becomes

$$\begin{gathered} p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = \\ = p(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) \times \\ \times p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x). \end{gathered}$$

Using a similar decomposition to that in Section 2.3.1, we have

$$p(e_t \mid b_t, a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x), \tag{3.1}$$

and

$$p(b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2}), B_x) = p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x). \tag{3.2}$$

We should note that, when B_x is a robotic agent, the environment model (3.1) would actually be $p(e_t \mid e_{t-1}, e_{t-2}, B_x)$, as the agent's action does not affect the environment.

Again, we deal with (3.2) through a model averaging problem for each agent B_x:

$$\begin{gathered} p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) = \\ = p(M_1|B_x)\ p(b_t \mid b_{t-1}, b_{t-2}, B_x) + p(M_2|B_x)\ p(b_t \mid a_t, B_x), \end{gathered}$$

where $p(M_i|B_x)$ denotes the probability that our agent gives to model M_i, assuming that the adversary is B_x, with $p(M_1|B_x) + p(M_2|B_x) = 1,\ p(M_i|B_x) \geq 0$. Finally, we use model averaging, see Clyde & George (2004) and Hoeting *et al.* (1999), over users, defined through

$$\begin{gathered} p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \\ = \sum_{B_x} \Big[p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) \times p(B_x) \Big]. \end{gathered} \tag{3.3}$$

The core of *classical conditioning* and *adversary's* models remains as in Chapter

2. In the implementation of our basic model, see Section 2.4.2, we use two matrix-beta models to store the corresponding data, an $n \times m$ matrix for the classical conditioning model and an $n \times n \times n$ matrix for the adversary's model, as $\mathcal{A}$ (for our agent) had m elements and $\mathcal{B}$ (for the user) had n. As the robot now faces users and agents, the size of the data structures would be different depending on the set of actions that the adversary, the agent believes is dealing with, may use. Specifically, in case our agent is facing another agent, it will store the classical conditioning model data into an $m \times m$ matrix and the adversary model data into an $m \times m \times m$ three-dimensional matrix.

We provide now some details about the preference model. As described in Section 2.4.3, each agent aims at satisfying five objectives. Since our agent may behave cooperatively or competitively, the preference model would include an additional component utility function at the fifth level of the objectives pyramid, as shown in Fig. 3.4, in relation with its social behavior.

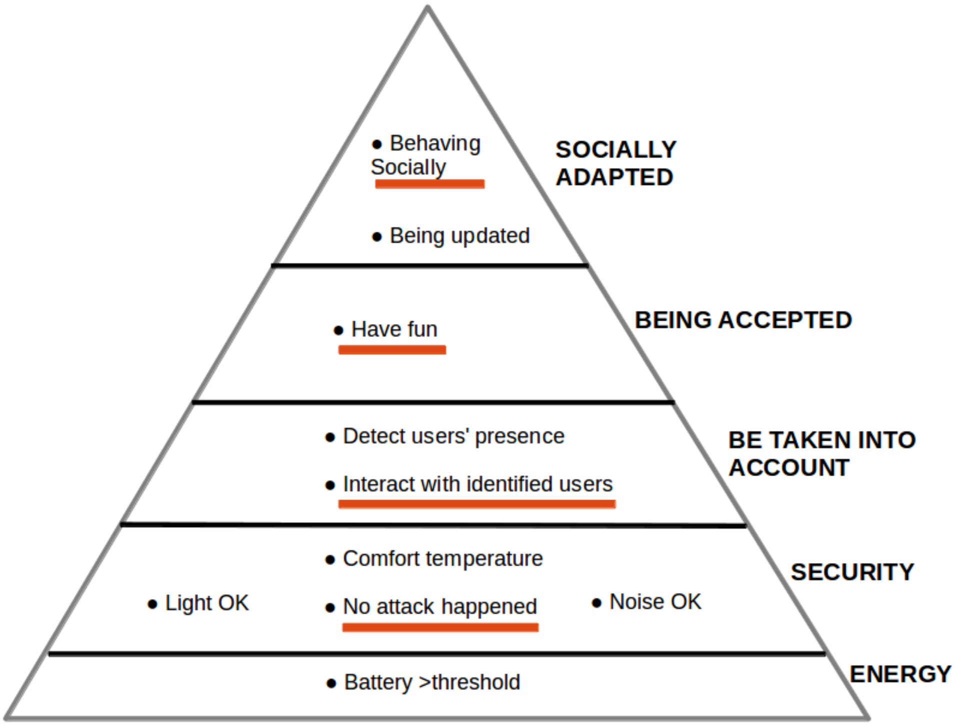


Figure 3.4: Pyramid of objectives.

The first objective $u_1(energy)$, as well as some component utility functions

(u_{22}(*temperature*), u_{23}(*light*), u_{24}(*noise*), u_{315}(*being touched*), u_{32}(*detection*) and u_{52}(*being updated*)), remains unvaried as none of them depends on interactions with other participants. For additional details, see Section 2.4.3. The other subobjectives, see those underlined in red in Fig. 3.4, will be extended because of the need of facing several users and agents and cope with social interactions. Specifically we have,

$$u_{21}(attack) = \begin{cases} 1, & \text{if no attack from any user or agent is inferred at } t \\ & \text{or at } t-1, \\ 0.5, & \text{if after an attack at } t-1, \text{ there was no attack at } t, \\ 0, & \text{otherwise,} \end{cases}$$

where attack refers to actions b_5=*attack*, b_6=*offend* and a_{13}=*offend*.

$$u_{311}(not\ ignored) = \begin{cases} 0, & \text{if the agent is ignored by any user at } t \\ 0.5, & \text{if it was ignored at } t-1, \text{ but was not at } t \\ 1, & \text{otherwise,} \end{cases}$$

$$u_{312}(being\ spoken) = \begin{cases} 1, & \text{if a grammar has been initiated at } t \text{ by a user} \\ & \text{or by another agent,} \\ 0.5, & \text{if a grammar was initiated at } t-1 \text{ by a user} \\ & \text{or by another agent,} \\ 0, & \text{otherwise,} \end{cases}$$

where *being spoken* refers to detecting a request to speak from a user (b_{10}=*speak*), starting a standard issues dialogue grammar, or a request for speaking by another agent (a_6=*speak*).

$$u_{313}(asked\ to\ play) = \begin{cases} 1, & \text{if the robot is asked to play by the user} \\ & \text{or by another agent at } t, \\ 0.5, & \text{if the robot was asked to play at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

where *asked to play* refers to detecting a request to play from the user (b_{13}= *play*), including the game's title, or a request for playing by another agent (a_7= *ask for playing*).

$$u_{314}(being\,ordered) = \begin{cases} 1, & \text{if the robot receives an order from any user at } t, \\ 0.5, & \text{if the robot received an order at } t-1 \text{ but not at } t, \\ 0, & \text{otherwise,} \end{cases}$$

where *being ordered* consists of detecting an order among a catalogue of actions within a certain grammar (b_{12}= *order*). *Being ordered, asked to play* or *being spoken* are evaluated through an ASR algorithm and the ChatScript modules, see Section 1.1 for an explanation, so they depend on the defined grammar and are detected in a probabilistic way.

$$u_{41}(play) = \begin{cases} 1, & \text{if the robot inferred a user or another agent} \\ & \text{playing around at } t, \\ 0.5, & \text{if the robot was playing with somebody at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

where *playing around* is referred to actions b_{13}=*play* and a_5=*play*, respectively.

$$u_{42}(flatter) = \begin{cases} 1, & \text{if the robot is flattered by a user} \\ & \text{or by another agent at } t, \\ 0.5, & \text{if the robot was flattered by a user} \\ & \text{or by another agent at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

being b_3=*flatter* and a_{12}=*flatter*, the incumbent actions.

$$u_{43}(stroke) = \begin{cases} 1, & \text{if the robot receives a stroke from a user at } t, \\ 0.5, & \text{if the robot received a stroke from a user at } t-1, \\ 0, & \text{otherwise,} \end{cases}$$

$$u_{44}(apologize) = \begin{cases} 1, & \text{if the robot receives an apology from a user} \\ & \text{or an agent after an attack,} \\ 0.5, & \text{if the robot received an apology from a user} \\ & \text{or an agent at } t-1 \text{ after an attack,} \\ 0, & \text{otherwise,} \end{cases}$$

being b_4=*apologize* and a_{14}=*apologize* the incumbent actions.

For its fifth objective, the robot checks whether it is being socially adapted. To do so, it evaluates whether it is considered as socially useful by its peers, and whether it has been updated recently. We represent this through

$$u_5(social\ adaptation) = w_{51} \times u_{51}(socially\ useful) + w_{52} \times u_{52}(being\ updated),$$

with $\sum_{i=1}^{2} w_{5i} = 1$, and weights ordered in importance as follows: $w_{51} >> w_{52} > 0$. Weights may be initialized as $w_{51} = 0.7,\quad$ and $w_{52} = 0.3$. The aim of component u_{51} is two-folded: on one hand, we want to evaluate whether the agent is somehow recognized as a member of the society, inferring the reactiveness of its opponents to its actions; on the other, we want to measure how good the reaction of its opponents is. Our implementation of these ideas will be

$$u_{51}(socially\ useful) = \frac{\sum_x p(M_2 \mid B_x)}{q} \times inter,$$

where $p(M_2 \mid B_x)$ is the probability used to represent the agent's estimation of how reactive the q opponents B_x, were, see (2.4); and *inter*, is the positive or negative impact of the reaction of the opponent B_x at t, with

$$inter = \begin{cases} 1, & \text{if } b_t \in \text{ affective actions,} \\ 0.5, & \text{if } b_t \notin \text{ affective actions and } \notin \text{aggresive actions,} \\ 0, & \text{otherwise,} \end{cases}$$

where the set of affective and aggresive actions are defined in Section 2.4.1.

The agent maximizes expected utility, which averages conditioning over the identified adversaries, thus taking into account model uncertainty, see Clyde &

George (2004) and Hoeting *et al.* (1999). Specifically, the agent solves

$$\max_{a_t \in \mathcal{A}} \psi(a_t) = \sum_{B_x} \left[\int \int u(a_t, b_t, e_t) \times \right.$$
$$\left. \times p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) db_t, de_t \right] \times p(B_x).$$

When one of the identification probabilities is much higher than the others for a certain adversary B_x^*, we could solve the simpler problem

$$\max_{a_t \in \mathcal{A}} \psi(a_t) = \int \int u(a_t, b_t, e_t) \times$$
$$\times p(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x^*) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x^*) db_t, de_t.$$

3.3 Supporting a competitive agent

We assume now that several agents compete to accomplish a certain goal, involving users in the scene, see Fig. 3.5. As an example, consider a case in which

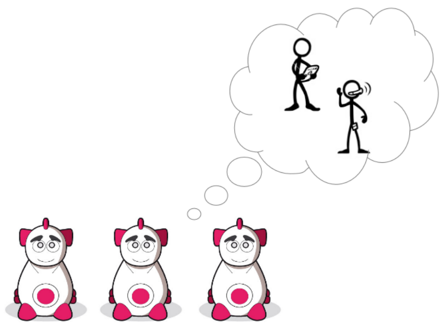


Figure 3.5: Several agents competing to satisfy various users.

there are three robots and two kids in a scene. The kids want to play a game, but would like to have one more player to do so. All agents want to play with the kids to satisfy its higher objectives, see Section 3.2.1, but just one of them will do it. The agents would compete to be chosen as the third player, being nicer, funnier or whatever, in order to be selected.

For comparison, these situations have been conceptualized through computing Nash Equilibria (NE) and under the ARA framework. We shall assume that there is communication among the agents, this being required for the NE case. Moreover, under the NE framework, we shall assume that there is a Computerised Trusted Third Party (CTTP) that would handle conflicts, computing the NE when needed. This may be an external system or one of the robotic agents that could adopt such role of trusted third party. There are certain conditions that the CTTP must take into account to compute the corresponding Nash Equilibria, see Aliprantis & Chakrabarti (2010):

- Agents aim at maximizing their expected payoff.
- Agents are perfectly rational and do not make mistakes in their execution.
- Agents are intelligent enough to deduce the solution of the game. As a consequence of this, they all know the equilibrium strategy to be played by the rest of the agents.
- Agents consider that deviating from their own strategy will not cause deviations by any other agent.
- It is common knowledge that all agents meet these conditions.

Working under ARA, we shall not make common knowledge assumptions. In case we need it for other purposes, we may allow agents to communicate.

There will be two different types of communication: among the participating robotic agents and each of them with the CTTP. The agents would be periodically transferring information to interact with each other. Whenever a conflict arises, the agents would send their beliefs, matrices and parameters, as well as their utilities to the CTTP, who would compute the required solution and send the corresponding strategies back to each participating agent.

Model

We have a set of r agents $\{A_1, A_2, \dots, A_r\}$ under a competing attitude, possibly in presence of a set of q users $\{B_1, B_2, \dots, B_q\}$, within an environment E. Collectively, at each decision period t they jointly implement their respective actions $a_t = (a_{1_t},\ a_{2_t} \dots a_{r_t})$, each of them in $\mathcal{A}$, whereas users will perform the corresponding actions $b_t = (b_{1_t},\ b_{2_t} \dots b_{q_t})$, all of them in $\mathcal{B}$.

The multiattribute utilities that the agents will obtain will be, respectively:

$$u_1(a_t, b_t, e_t), \quad u_2(a_t, b_t, e_t), \quad \dots \ , \ u_r(a_t, b_t, e_t).$$

The forecasting model for agent A_1, which we assume will be the supported one, will be of the form

$$p_1(e_t, b_t, a_{-1_t} \mid a_{1_t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})), \tag{3.4}$$

where a_{-1_t} would be the actions of all agents performed at time t, excluding our supported agent action, a_{1_t}. As before, we assume that e_t remains exclusively under the users' control. Then, (3.4) becomes

$$p_1(e_t \mid b_t, e_{t-1}, e_{t-2}) \times p_1(b_t, a_{-1_t} \mid a_{1_t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})).$$

We assume that when our agent A_1 faces another agent, the forecasted action of that robotic agent will depend only on its previous actions and on the action of agent A_1. The users' actions will depend on all the agent's actions. Then, equation (3.4) becomes:

$$p_1(e_t \mid b_t, e_{t-1}, e_{t-2}) \times \prod_{j=2}^{r} p_1(a_{j_t} \mid a_{j_{t-1}}, a_{j_{t-2}}, a_{1_{t-1}}) \times \prod_{j=1}^{q} p_1(b_{j_t} \mid a_t, b_{j_{t-1}}, b_{j_{t-2}}). \tag{3.5}$$

Thus, the components of the forecasting models for agent A_1 are: the first term of (3.5), the *environmental model*, and the rest, which is the model to forecast the adversaries' actions. This will be decomposed into the *adversary* and the *classical* *conditioning* models, similarly to (3.3):

$$p_1(e_t, b_t, a_{-1_t} \mid a_{1_t}, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \tag{3.6}$$
$$= \sum_{B_x} \Bigg[p_1(e_t \mid b_t, e_{t-1}, e_{t-2}, B_x) \times p_1(b_t \mid a_t, b_{t-1}, b_{t-2}, B_x) \times$$
$$\times p_1(a_{-1_t} \mid a_{1_t}, a_{-1_{t-1}}, a_{-1_{t-2}}, B_x) \times p(B_x) \Bigg].$$

They are combined through model averaging techniques. Forecasting other agents' actions will be defined as forecasting the user's actions in Section 3.2.1, evaluating the evolution of its own behavior and how reactive they are to agent A_1's actions. We have:

$$p_1(a_{j_t} \mid a_{1_{t-1}}, a_{j_{t-1}}, a_{j_{t-2}}) = \tag{3.7}$$
$$= p(M_1^j) p_1(a_{j_t} \mid a_{j_{t-1}}, a_{j_{t-2}}) + p(M_2^j) p_1(a_{j_t} \mid a_{1_{t-1}}),$$

with $\sum_i p(M_i^j) = 1,\ p(M_i^j) \geq 0$, for each agent $j \neq 1$. Forecasting the users' actions would be extended to include the reaction of the user to the actions of every agent, hence:

$$p_1(b_{k_t} \mid a_t, b_{k_{t-1}} b_{k_{t-2}}) = p(M_0^{B_k}) p_1(b_{k_t} \mid b_{k_{t-1}}, b_{k_{t-2}}) + \sum_{j=1}^{r} p(M_j^{B_k}) p_1(b_{k_t} \mid a_{j_t}), \tag{3.8}$$

with $\sum_i p(M_i^{B_k}) = 1,\ p(M_i^{B_k}) \geq 0$, for each user $k \in Q$.

We may use matrix-beta models with prior and posterior Dirichlet distributions over the adversary and the classical conditioning models in (3.7) and (3.8). For the adversary models we have, for each user b_k, a posterior Dirichlet distribution

$$p(b_{k_t} \mid b_{k_{t-1}} = j, b_{k_{t-2}} = i, D_t) \sim Dir(\rho_{ij1} + h_{ij1}, \ldots, \rho_{ijn} + h_{ijn}),\ b_{k_t} \in \{b_1, b_2, ..., b_n\}, \tag{3.9}$$

where h_{ijk} designates the number of occurrences when the opponent did $b_{k_t} = b_k$ after having done $b_{k_{t-1}} = b_j$ and $b_{k_{t-2}} = b_i$; and ρ_{ijk} are the prior parameters with

$\rho_{ijk} \geq 0$ for $i, j, k = 1, \ldots, n$. In case our adversary is an agent, for each agent a_j,

$$p(a_{j_t} \mid a_{j_{t-1}} = j, a_{j_{t-2}} = i, D_t) \sim Dir(\rho_{ij1} + h_{ij1}, \ldots, \rho_{ijm} + h_{ijm}),\ a_{j_t} \in \{a_1, a_2, ..., a_m\}, \tag{3.10}$$

where h_{ijk} designates the number of occurrences when the opponent did $a_{j_t} = a_k$ after having done $a_{j_{t-1}} = a_j$ and $a_{j_{t-2}} = a_i$; and ρ_{ijk} are the prior parameters with $\rho_{ijk} \geq 0$ for $i, j, k = 1, \ldots, m$. The required data will be stored in a three-dimensional matrix, where the last row accumulates the sum of row values for each column and each layer.

For the classical conditioning models, we have, for each human opponent b_{k_t}, a posterior Dirichlet distribution

$$p(b_{k_t} \mid a_{j_t} = a_j, D_t) \sim Dir(\beta_{1j} + h_{1j}, \ldots, \beta_{nj} + h_{nj}),\ b_{k_t} \in \{b_1, b_2, ..., b_n\}, \tag{3.11}$$

where, similarly, h_{ij} designates the number of occurrences when the opponent did $b_{k_t} = b_i$ after observing an agent j doing $a_{j_t} = a_j$; and β_{ij} are the prior parameters with $\beta_{ij} \geq 0$ for $i = 1, \ldots, n$. In case we are facing an agent j, for each agent a_k,

$$p(a_{j_t} \mid a_{k_t} = j, D_t) \sim Dir(\beta_{1j} + h_{1j}, \ldots, \beta_{mj} + h_{mj}),\ a_{j_t} \in \{a_1, a_2, ..., a_m\}, \tag{3.12}$$

where, similarly, h_{ij} designates the number of occurrences when the opponent did $a_{j_t} = a_i$ after observing another agent doing $a_{k_t} = a_j$; and β_{ij} are the prior parameters with $\beta_{ij} \geq 0$ for $i = 1, \ldots, m$. In this case, the required data will be stored in a two-dimensional matrix.

Computing Nash Equilibria Each agent aims at maximizing its expected utility. Under the standard game-theoretic approach, the agent's utilities and probabilities are common knowledge. When the agents implement a_t, the j-th agent expected utility will be:

$$\psi_j(a_t) = \int \ldots \int u_j(a_t, b_t, e_t) \times \tag{3.13}$$
$$\times \left[\prod_{k=1}^{q} p_j(b_{k_t} \mid a_t, b_{k_{t-1}}, b_{k_{t-2}}) \times p_j(e_t \mid b_{k_t}, e_{t-1}, e_{t-2}) \right] \; db_{1_t} \; \ldots \; db_{q_t} de_t.$$

As we pointed out, we assume that a CTTP would solve the conflicts, and there will be communication among the agents.

In order to compute the Nash Equilibria, the agents may adopt mixed strategies. Suppose that $v = (v_1, \ldots, v_r) \in V$ are the corresponding mixed strategies for agents $A_1, \ldots, A_r$, respectively, with $v_j = (v_{j_1}, \ldots, v_{j_m})$, $\sum_{i=1}^{m} v_{j_i} = 1$, $v_{j_i} \geq 0, \forall a_i \in \mathcal{A}$, for each agent action $a_i \in \mathcal{A}$, being $i = 1, \ldots, m$ and each agent j. Then, the CTTP would compute, for each agent,

$$\psi_j(v) = \int \ldots \int \sum_{i_1=1}^{m} \ldots \sum_{i_r=1}^{m} v_{1_{i_1}} \ldots v_{r_{i_r}} u_j(a_t, b_t, e_t) \times$$
$$\times \left[\prod_{k=1}^{q} p_j(b_{k_t} \mid a_t, b_{k_{t-1}}, b_{k_{t-2}}) p_j(e_t \mid b_{k_t}, e_{t-1}, e_{t-2}) \right] db_{1_t} \ldots db_{q_t} de_t.$$

The CTTP would compute the Nash Equilibria based on the following system of equations:

$$\frac{\partial \psi_1(v)}{\partial v_{1_i}} = 0, \; \frac{\partial \psi_2(v)}{\partial v_{2_i}} = 0, \; \ldots \; , \frac{\partial \psi_r(v)}{\partial v_{r_i}} = 0,$$
$$\frac{\partial^2 \psi_1(v)}{\partial^2 v_{1_i}} < 0, \; \frac{\partial^2 \psi_2(v)}{\partial^2 v_{2_i}} < 0, \; \ldots \; , \frac{\partial^2 \psi_r(v)}{\partial^2 v_{r_i}} < 0,$$

for each agent action $a_i \in \mathcal{A}$, being $i = 1, \ldots, m$. Any solution $(v_1, \ldots, v_r)$ of the above system, with $v_j \geq 0$ and $\sum_j^r v_j = 1$, is a Nash equilibrium $(v_1^*, v_2^*, \ldots, v_r^*)$, see Aliprantis & Chakrabarti (2010). There could be several equilibria with no unambiguous criteria to further discern among them, see Raiffa *et al.* (2007). For general descriptions on computing Nash Equilibria, see Nisan *et al.* (2007).

ARA Solving agents We consider now the problem from the ARA point of view. In this case, as we said, neither communication, nor common knowledge are required. We are supporting one of the agents (A_1), to make a decision facing several users and other agents. The agent will aim at maximizing its expected utility based on forecasts of the other agents defined through

$$\max_{a_{1_t}} \psi_1(a_{1_t}) = \int \ldots \int \psi_1(a_t) \left[\prod_{j=2}^{r} p_1(a_{j_t} \mid a_{1_{t-1}}, a_{j_{t-1}}, a_{j_{t-2}}) \right] da_{2_t} \ldots da_{r_t},$$

where the relevant probability models were described in (3.9), (3.10), (3.11) and (3.12). As our agent considers its adversaries as level-0 opponents, we will be at level-1 in the ARA hierarchy.

3.3.1 A second competitive scenario

In this case, each agent is interacting with its own user, supporting her within a competition against other user-agent teams, see Fig. 3.6. As an example, consider a case in which three teams are involved: couples robot A_1 - child B_1; robot A_2 - child B_2; and, finally, robot A_3 - child B_3. Each of them works on school assignments aiming at getting the highest grade. Each agent shall support its own user in making decisions, forecasting what the other agents would do. Assumptions similar to those made earlier in this Section will be made here.

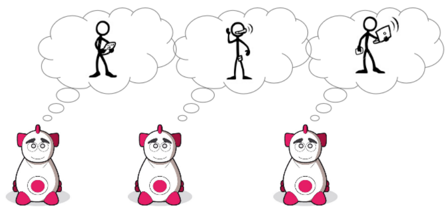


Figure 3.6: Several user-agent teams competing.

Model

We have a set of r agents $\{A_1, A_2, \ldots, A_r\}$ under a competing attitude, supporting simultaneously their corresponding users within an environment. Collectively, at each decision period t they jointly implement their respective actions $a_t = (a_{1_t},\ a_{2_t} \dots a_{r_t})$, as before, whereas the user that interacts with the corresponding agent will perform action b_{j_t}, being j the index of the incumbent agent. The multi-attribute utilities that the agents will obtain will be, respectively:

$$u_1(a_t, b_{1_t}, e_t), \quad u_2(a_t, b_{2_t}, e_t), \quad \dots \ , \ u_r(a_t, b_{r_t}, e_t).$$

The forecasting model for agent A_1, which teams with user B_1, would be

$$p_1(e_t, b_{1_t}, a_{-1_t} \mid a_{1_t}, (e_{t-1}, a_{t-1}, b_{1_{t-1}}), (e_{t-2}, a_{t-2}, b_{1_{t-2}})), \tag{3.14}$$

with a_{-1_t} as before. Simplifications and assumptions related to the forecasting models would be analogous to those of Section 3.3. Equation (3.14) ends up decomposed in $p_1(e_t \mid b_{1_t}, e_{t-1}, e_{t-2})$, the *environmental model*, and

$$p_1(a_{-1_t} \mid a_{1_{t-1}}, a_{-1_{t-1}}, a_{-1_{t-2}}) \times p_1(b_{1_t} \mid a_t, b_{1_{t-1}}, b_{1_{t-2}}),$$

which will be, respectively, decomposed in the *adversary* and the *classical conditioning* models, as we did in the previous scenario, then combined, through model averaging techniques as in (3.6).

ARA Solving agents

From an ARA perspective, we are supporting one of the agents (A_1) facing its own user (B_1) and other agents (A_2, ... ,A_r). It will maximize its expected utility, defined through

$$\max_{a_{1_t}} \psi_1(a_{1_t}) = \int \dots \int \psi_1(a_t) \left[\prod_{j=2}^{r} p_1(a_{j_t} \mid a_{1_{t-1}}, a_{j_{t-1}}, a_{j_{t-2}}) \right] da_{2_t} \dots da_{r_t},$$

where the relevant probability models are similar to those described in Section 3.3.

3.4 Supporting cooperative agents

So far, we have described how competitive agents could behave. We focus now on cooperative cases, in which several agents collaborate to find out the best solution that satisfy them to help users in achieving a specific task, as suggested in Fig. 3.7. As an example, consider three robots that want to support three kids in their weekly school assignments. The robots are under a cooperative attitude, so that they would look for helping the children together, replicating a collaborative work environment at school. We provide a solution for these scenarios introducing a solution concept for cooperative games.

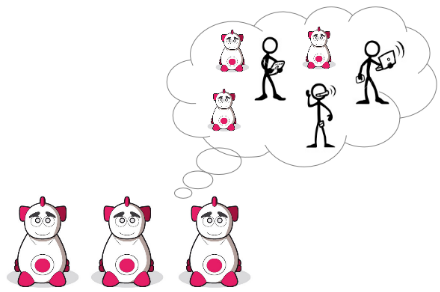


Figure 3.7: Society of cooperative agents.

Model

We have a set of r agents $\{A_1, A_2, \ldots, A_r\}$ under a cooperative attitude, possibly in presence of a set of q users $\{B_1, B_2, \ldots, B_q\}$, within an environment E. As in Section 3.3, at each planning period t, collectively, they jointly implement their respective actions $a_t = (a_{1_t},\ a_{2_t}, \ldots, a_{r_t})$, each of them in $\mathcal{A}_j$, $j = 1, ..., r$, whereas the users will perform their corresponding actions $b_t = (b_{1_t},\ b_{2_t}, \ldots, b_{q_t})$, all of them in $\mathcal{B}$. The multiattribute utilities that the agents will obtain will be, respectively:

$$u_1(a_t, b_t, e_t), \quad u_2(a_t, b_t, e_t), \quad \ldots \ , \ u_r(a_t, b_t, e_t),$$

thus depending not only on what they do, but also on what the other agents and users perform, as well as the environmental state.

As we did in Section 3.3, we assume that a CTTP would play the role of an arbitrator solving, in this case, the cooperative game. There will be communication among the agents and with the arbitrator, sending, each agent, its beliefs, matrices and parameters, and utilities as required. It is common knowledge that each agent individually aims at maximizing its expected utility as in (3.13).

Once the CTTP has computed ψ_j, for each agent $j \in R$, we may use cooperative game theory, to find the solution within this scenario. There are different methods to solve such situations within the game theoretic framework, see Thomson (1994) and Thomson (2009) for reviews. We shall use a method that finds a solution maximizing the distance from the ARA solutions, a non-cooperative Nash equilibria or, more generally, to a disagreement point.

A cooperative game is defined by the tuple $(F,\ d)$, where F is the set of (expected) utilities attainable by the agents, in our case $F = \{x \in R^r : x = (\psi_1(a), ..., \psi_r(a)),$ for $a \in \mathcal{A}^r\}$, thus being finite. $d = (d_1, \ldots, d_r)$ is the disagreement point, i.e. the prespecified utilities obtained when there is no agreement among the players. If, otherwise, an agreement is reached, the alternative chosen is the solution concept of the game, defined by $\phi_j(F, d)$, for each agent j. As mentioned F here is finite, in contrast of traditional bargaining theory which tends to assume that F is a convex and d-comprehensive set, that is if $x \in F$ and $x \geq z \geq d$, then $z \in F$.Given the disagreement point d, we shall look for the point $x \in F$, where $x \geq d$, which maximizes the distance function for each agent $j \in R$, being x_j the utility obtained by agent j; and d_j the utility corresponding to agent j if the disagreement point is reached. To define our distance, we stem from by the classic cooperative game solution due to Yu (1973), which looks for the solution minimizing an L^p distance to an ideal point $\hat{x}$, where $\hat{x}_j = \arg\max_{x \in F} x_j$ for each player j and $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_r)$, with the parameter $1 \leq p \leq \infty$ being used as "balancing factor". Yu proposes looking for the point $x \in F$, which minimizes

$$D_p(x, \hat{x}) = \left[\sum_{j=1}^{r} (\hat{x}_j - x_j)^p \right]^{\frac{1}{p}}.$$

Based on these ideas, and considering only solutions $x \in F$ satisfying $x \geq d$, we shall use the following maximum distance problem to find our cooperative solution

$$\phi(F,d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} D_p(x,d) = \arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \left[\sum_{j=1}^{r} (x_j - d_j)^p \right]^{1/p}.$$

Note that when $p = 1$, the optimization problem is equivalent to

$$\arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \sum_{j=1}^{r} x_j,$$

which corresponds to the utilitarian solution, see Thomson (1981). When, $p = \infty$, the optimization problem is

$$\arg \max_{j \in R} \max\ (x_j - d_j),$$

thus aiming at maximizing the maximum payoff.

Assessing the cooperative solution

We now assess our cooperative solution. To do so, we shall evaluate which standard cooperative game theoretic axioms does it fulfill. We have adopted the axioms in Wu (2007), who axiomatizes several bargaining solution concepts, including Nash and Kalai-Smorodinsky solutions, over finite sets. Specifically, we shall evaluate which axioms characterizing such solutions are fulfilled by our proposal. We first briefly describe the intuition behind each axiom, and then either prove that the proposed solution concept satisfies it, or provide a counterexample. For illustrative purposes, figures and counterexamples will be based on two-player games. In the provided figures, from 3.8 to 3.13, dark dots refer to alternatives belonging to the feasible set F; white dots belong to a different feasible set, G, to be defined in each case; d is the disagreement point and x^* is the solution of the game. The maximum distance from the disagreement point will be represented by a dotted line.

The efficiency axiom requires that the solution lies in the Pareto frontier: there is no alternative in the feasible set, such that, a deviation from the solution to that alternative can make a player better off, without making other players worse off, see Fig. 3.8.

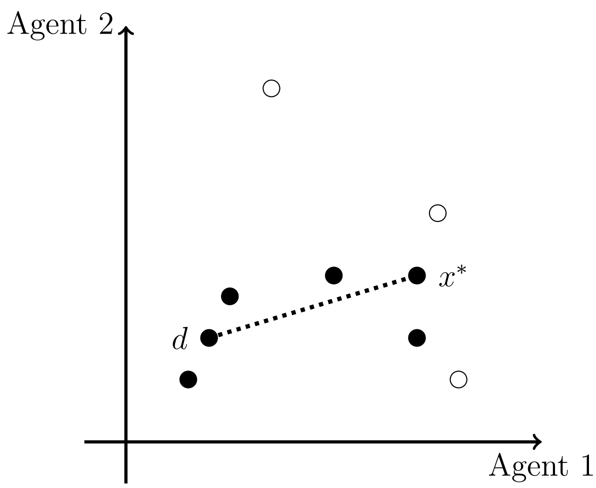


Figure 3.8: An alternative that improves x^* is out of F.

Axiom 1 Efficiency (E): *For any $x^* \in \phi(F, d)$, there is no $z \in F$, with $z \geq d$, such that $z > x^*$.*

Proof Suppose that $z \in F$, with $z \geq d$, that dominates x^*. Thus, $z_j \geq x_j^*$, $\forall j$ with at least an strict inequality for some j. Then, $(z_j - d_j)^p \geq (x_j^* - d_j)^p, \forall j$ with at least an strict inequality for some j. Therefore, $\sum_j^r (z_j - d_j)^p > \sum_j^r (x_j^* - d_j)^p$ and $D_p(z, d) > D_p(x^*, d)$, which is a contradiction. ■

Weak efficiency is less restrictive than **E**. It requires that there is no outcome in the feasible set strictly better than the solution for all agents. Since our concept satisfies efficiency, it also satisfies weak efficiency.

Axiom 2 Weak Efficiency (WE): *For any $x^* \in \phi(F, d)$, there is no $z \in F$ such that $z >> x^*$.*

The next axiom requires that in symmetric problems, for each cooperative solution, its symmetric is also a solution, see Fig. 3.9.

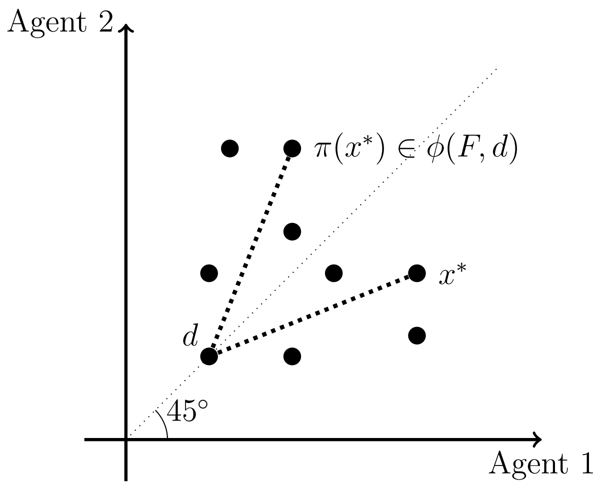


Figure 3.9: If (F, d) is symmetric, the solution will be symmetric.

Axiom 3 Symmetry (SYM): *If F is symmetric with respect to the diagonal, and the disagreement point is also symmetric ($d_i = d_j, \forall i, j \in R$), then $x^* \in \phi(F, d)$, $\pi(x^*) \in \phi(F, d)$ for all permutations π.*

Proof If F is symmetric with respect to the diagonal, we have that for each alternative $x^* \in \phi(F, d)$, $D_p(x^*, d) = D_p(\pi(x^*), d)$ since

$$D_p(x^*, d)^p = \sum_{j=1}^{r} (x_j^* - d_j)^p = \sum_{j=1}^{r} \left(\pi(x_j^*) - \pi(d_j)\right)^p =$$

$$= \sum_{j=1}^{r} \left(\pi(x_j^*) - d_j\right)^p = D_p(\pi(x^*), d)^p.$$

Then, as $D_p(x^*, d) \geq D_p(x, d)$ for any $x \in F$, $\pi(x^*) \in \phi(F, d)$. ■

Utility is invariant with respect to positive affine transformations, see von Neumann & Morgenstern (1944). Note that each player can use a different scale to measure utility, implying that there is no interpersonal comparison of utilities, see Fig. 3.10. Scale independence holds when agents use the same scale transformation, but not otherwise.

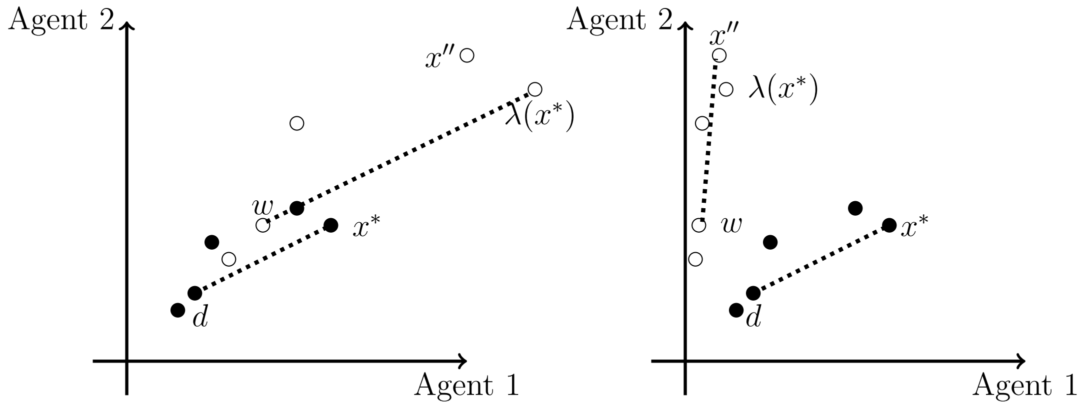


Figure 3.10: Under different players' scales, the solution changes.

Axiom 4 Scale Independence (SI): *For any numbers $\lambda_j > 0$ and β_j for each $j \in R$, if $G = \{(\lambda_j x_j + \beta_j) : x_j \in F\}$ and $w = (\lambda_j d_j + \beta_j)$, then $\phi(G, w) = (\lambda_j \phi_j(F, d) + \beta_j)$.*

Proof The result only holds when $\lambda_i = \lambda_j\ \forall i, j \in R$. Note that in such case, if x' designates the transformed solution, we have

$$D_p(x', w)^p = \sum_{j=1}^{r} ((\lambda x_j^* + \beta_j) - (\lambda d_j + \beta_j))^p = \lambda^p D_p(x, d)^p,$$

from which the result immediately follows. ■

As a counterexample, consider a two-player game with $F = \{x^1 = (0.4, 0.65), x^2 = (0.45, 0.5)\}$, and disagreement point $d = (0.1, 0.1)$. Suppose that $p = 1$. The solution of this game is x^1, as $D_1(x^1, d) > D_1(x^2, d)$. Suppose that $\lambda_1 = 5,\ \lambda_2 = 1/3,\ \beta_1 = \beta_2 = 1$. We have that $x' = (\lambda_1 x_1^* + \beta_1,\ \lambda_2 x_2^* + \beta_2) = (5 \times 0.4 + 1,\ 1/3 \times 0.65 + 1)$. Then, $D_1(x', w) = 1.316$, whereas $D_1(x'' = (\lambda_1 x_1^2 + \beta_1,\ \lambda_2 x_2^2 + \beta_2), w) = 1.616$. Therefore, $D_1(x'', w) > D_1(x', w)$.

The domination axiom says that, if there is an alternative z in F such that z is greater than some outcome x^* in the solution, then z must be a part of the solution as well, see Fig. 3.11. Since our solution concept satisfies axiom **E**, it cannot satisfy this axiom **D**.

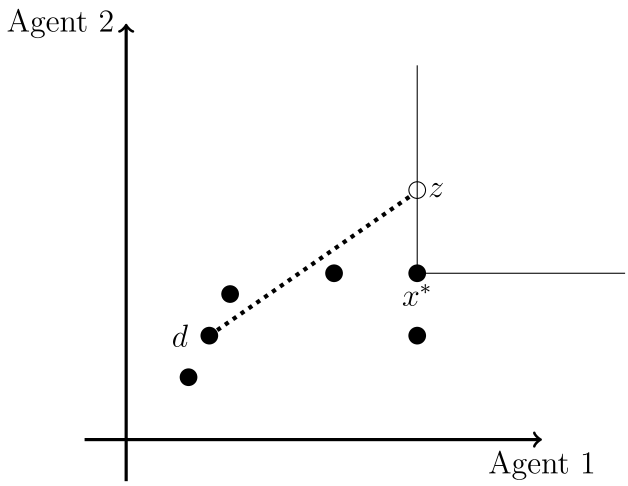


Figure 3.11: No alternative in F improves x^*.

Axiom 5 Domination (D): *For any alternative $z \in F$, if $z > x^*$ for some $x^* \in \phi(F, d)$, then $z \in \phi(F, d)$.*

The intuition underlying the next axiom is that an alternative which does not belong to the solution set of a problem, cannot be considered as a relevant solution candidate when the problem gets shrunk, see Fig. 3.12.

Axiom 6 Contraction Independence (CI): *For any feasible set G such that $F \subseteq G$ and $F \cap \phi(G, d) \neq \emptyset$, then $\phi(F, d) = F \cap \phi(G, d)$.*

Proof Let G be a set such that $F \subseteq G$ and $F \cap \phi(G, d) \neq \emptyset$. Suppose that $z \in \phi(F, d)$ and $z \notin \phi(G, d)$. Then, $D_p(z, d) \geq D_p(x, d)$, $\forall x \in F$, but there is $y \in G$ such that $D_p(y, d) > D_p(z, d)$. Let $s \in F \cap \phi(G, d)$, which exists due to our hypothesis. Then, $D_p(z, d) \geq D_p(s, d)$ and $D_p(y, d) = D_p(s, d) > D_p(z, d)$, which is a contradiction. Therefore, $\phi(F, d) \subseteq F \cap \phi(G, d)$. It is easy to prove that $F \cap \phi(G, d) \subseteq \phi(F, d)$. ∎

A less restrictive version of the **CI** axiom, is **WCI** below. In this case, Wu (2007) considers that the ideal point of both feasible sets must be the same. Since our concept satisfies CI, it does satisfy WCI.

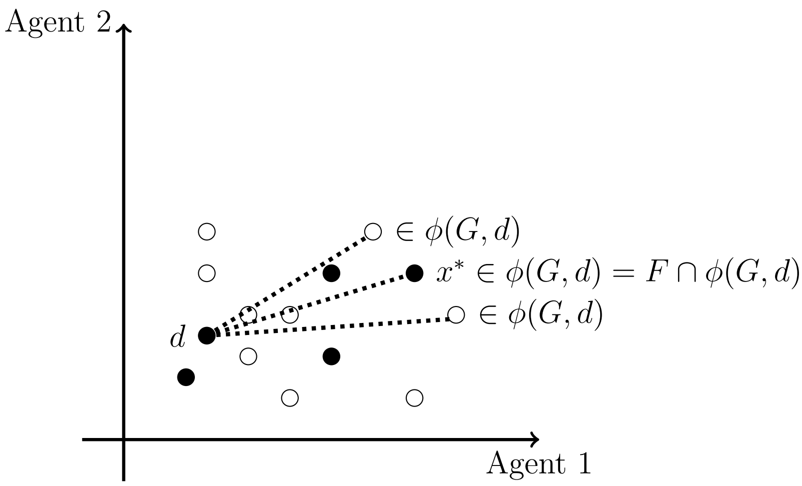


Figure 3.12: If G shrinks to F, $x^* \in \phi(G,d)$ would be the only candidate in F.

Axiom 7 Weak Contraction Independence (WCI): *For any feasible set G such that the ideal point in F is the same as in G ($\hat{x} = \hat{z}$, with $\hat{x} \in F$, and $\hat{z} \in G$), if $F \subseteq G$ and $F \cap \phi(G,d) \neq \emptyset$, then $\phi(F,d) = F \cap \phi(G,d)$.*

Finally, the intuition behind axiom **REC** below is that, when x^* is the unique solution to a symmetric problem with respect to the diagonal, when we enlarge the original problem to a larger one by adding other alternatives preserving symmetry, and there is no new alternative that strictly dominates x^*, we cannot exclude x^* from the solution of the new problem, see Fig. 3.13.

Axiom 8 Restricted Expansion Consistency (REC): *For any feasible set G such that $F \subseteq G$, if F and G are both symmetric, $\{x^*\} = \phi(F,d)$ and x^* is weakly efficient in G, then $x^* \in \phi(G,d)$.*

This axiom is not satisfied. Consider a two-player game with symmetric feasible set $F = \{x^1 = (0.7, 0.7),\ x^2 = (0.3, 0.5),\ x^3 = (0.5, 0.3)\}$, and disagreement point $d = (0.2, 0.2)$. We fix $p = 1$. x^1 is the unique solution of the game, since $D_1(x^1, d) = 1 > 0.4 = D_1(x^2, d) = D_1(x^3, d)$. Suppose that we enlarge F to the symmetric set $G = \{x^1 = (0.7, 0.7),\ x^2 = (0.3, 0.5),\ x^3 = (0.5, 0.3),\ x^4 = (0.7, 0.9),\ x^5 = (0.9, 0.7),\ x^6 = (0.6, 0.4),\ x^7 = (0.4, 0.6)\}$, with d as before. No

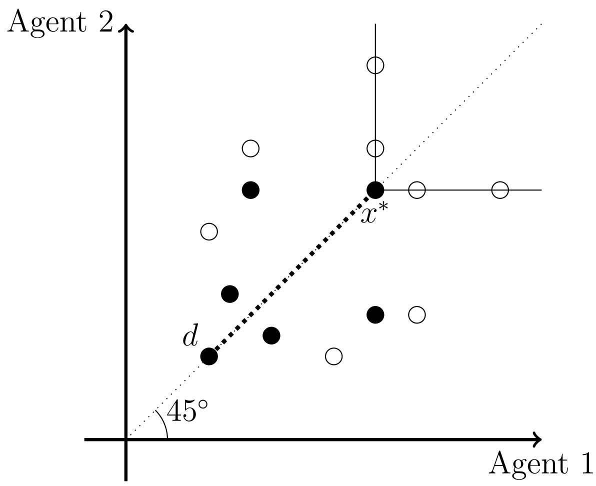


Figure 3.13: Enlarging F to G, being x^* weakly dominated, x^* could be in $\phi(G, d)$.

alternative in G strictly dominates the solution of F, which is x^1. However, note that $D_1(x^4, d) = D_1(x^5, d) = 1.2 > 1 = D_1(x^1, d)$, so that $x^1 \notin \phi(G, d)$.

Table 3.1 describes which axioms are satisfied and which ones are not by the Max D_p solution, Nash Bargaining solution and Kalai-Smorodinsky's. The Nash Bargaining solution for the domain of finite sets is the unique solution that satisfies axioms E, SYM, SI, and CI, see Wu (2007) and Mariotti (1998) for an alternative approach. The Kalai-Smorodinsky solution is the unique solution which satisfies SI, WE, WCI, REC and D, see Wu (2007). For an alternative extension of the Kalai-Smorodinsky solution to finite bargaining problems, see Nagahisa & Tanaka (2002) which focus on variations on the ideal point, while Wu (2007) focuses on expansion consistency, which we have considered more relevant to our case. Our solution concept satisfies E, SYM, CI, and, consequently, WE and WCI.

Axiom	Max D_p solution	Nash	K-S
E	✓	✓	✗
WE	(✓)	(✓)	✓
SYM	✓	✓	✗
SI	✗	✓	✓
D	✗	✗	✓
CI	✓	✓	✗
WCI	(✓)	(✓)	✓
REC	✗	✗	✓

Table 3.1: Summary table with the satisfied and unsatisfied axioms.

Note that since several alternatives may maximize the D_p distance. To come out with a single solution, we may randomize among the alternative optima. As an example, let us consider a two-player game with $F = \{x^1 = (0.5, 0.4),\ x^2 = (0.3, 0.6),\ x^3 = (0.5, 0.3)\}$, and disagreement point $d = (0.2, 0.2)$. x^1 and x^2 are the solutions of the maximum distance problem, since $D_1(x^1, d) = D_1(x^2, d) = 0.5 > 0.4 = D_1(x^3, d)$. We have no reason to prefer one of the solutions above the other. In order to be fair with all participants, we could choose randomly between both alternatives with equal probabilities.

3.5 From competition to cooperation

As described throughout this chapter, agents may compete or cooperate among them to reach their objectives over time. As we are dealing with autonomous agents we expect them to choose when to cooperate or compete with no user telling them what to do, forming an autonomuos society. This entails adopting some type of parametric model with parameters regulating the competitiveness and cooperativeness of the agents, possibly as described below.

For each agent j, we consider two non-negative parameters w_{j1} and w_{j2}, with $w_{j1}, w_{j2} \geq 0$ and $w_{j1} + w_{j2} = 1$, which shall allow them to modify their behavior. w_{j1} will describe the degree of cooperativeness of the agent, whereas w_{j2} will refer to its competitiveness. In both cases, the bigger the parameter, the higher such degree. As there is communication between the agents and the arbitrator, each agent j will submit its parameters as well as other required preference information to an arbitrator, which would compute an average value of those parameters to find the *society's attitude towards cooperation*, e.g. through

$$w_1 = \frac{1}{r} \sum_{j=1^r} w_{j1}, \;\; w_2 = \frac{1}{r} \sum_{j=1^r} w_{j2}.$$

Clearly, $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$.

We then propose the following solution concept

$$\phi(F, d) = \arg \max_{\substack{s.t.\; x \in F \\ x \geq d}} \Big(w_1 \; D_p(x, d) - w_2 \; D_q(x, d) \Big), \tag{3.15}$$

for given D_p and D_q distances, with $p \neq q$. Depending on those parameters, our proposed method shall allow the agents to modify their behavior maximizing, or minimizing, the distance from the disagreement point d among the feasible solutions ($x \in F$) dominating d. Then, the CTTP will compute the solution concept of the cooperative game, sending back the suggested agreement to the involved agents in the game.

Indeed, under a fully cooperative environment, $w_{j1} = 1$ for each agent j, the society will have parameters $w_1 = 1$ and $w_2 = 0$, and (3.15) becomes:

$$\phi(F, d) = \arg \max_{\substack{s.t.\; x \in F \\ x \geq d}} D_p(x, d),$$

corresponding to the solution concept in Section 3.4.

Similarly, under a fully competitive environment, $w_{j1} = 0$ for each agent j, the society parameters will be $w_1 = 0$ and $w_2 = 1$. The arbitrator would then solve

$$\phi(F, d) = \arg \max_{\substack{s.t.\; x \in F \\ x \geq d}} \Big(- \; \max D_q(x, d) \Big) = \arg \min_{\substack{s.t.\; x \in F \\ x \geq d}} D_q(x, d), \tag{3.16}$$

whose solution is the disagreement point d, which, as suggested, would be computed through ARA, corresponding to a non-cooperative solution concept.

We may then consider the behavior of the solution when the agents' attitudes are in between, with $0 \leq w_{j1} \leq 1$, $\forall j$. Then, $0 < w_1, w_2 < 1$. For illustrative purposes, in Fig. 3.14, we show how the corresponding objective function changes with the values of w_1 and w_2 for a two-player case, when $d = (0, 0)$, $p = 1$ and $q = \infty$, in which case we aim at solving

$$\arg \max_{\substack{s.t.\ x \in F \\ x \geq d}} \left(w_1 \sum_{j=1}^{r} (x_j - d_j) - w_2 \max_{j \in R} \{(x_j - d_j)\} \right).$$

In the figures, we show the isovalue functions for level 1 and the corresponding growth directions for various values of $w_1 \in [0, 1]$.

Representing the case of a fully cooperative society, with $w_1 = 1$ and $w_2 = 0$, our solution concept tries to maximize the sum of payoffs of both players, see Fig. 3.14(a). In the opposite case, with a fully competitive society, see (3.16), $w_1 = 0$ and $w_2 = 1$, the maximum payoff of each player shall be minimized, see Fig. 3.14(h). From Figs. 3.14(b) and 3.14(g), we reflect mixed behaviors as described in (3.15): e.g. Fig. 3.14(b) reflects the behavior when the society parameters are $w_1 = 0.8$ and $w_2 = 0.2$, and we may appreciate that improvement leads to more cooperative situation, whereas 3.14(g), with $w_1 = 0.2$ suggests improvement towards a more competitive situation.

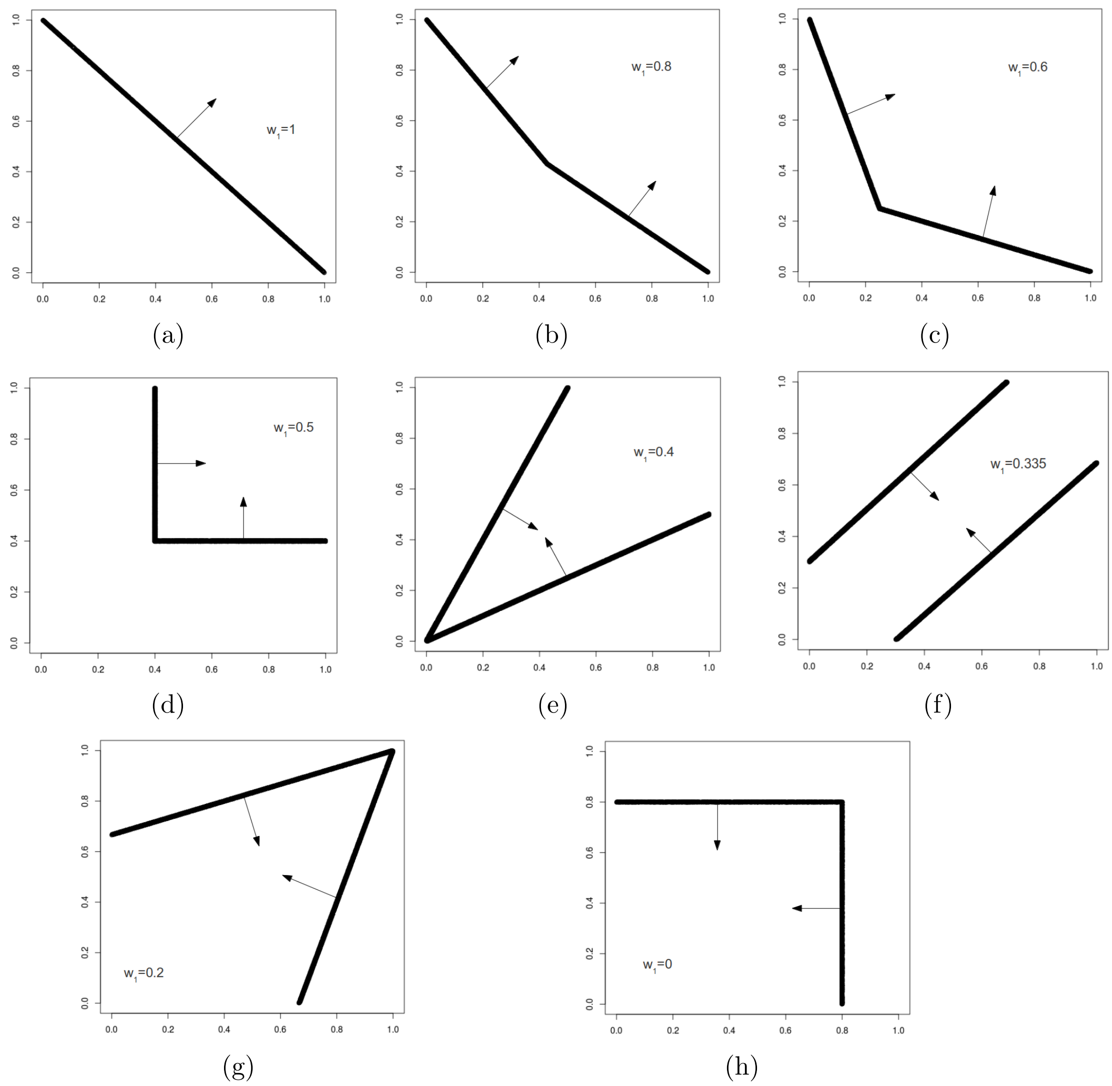
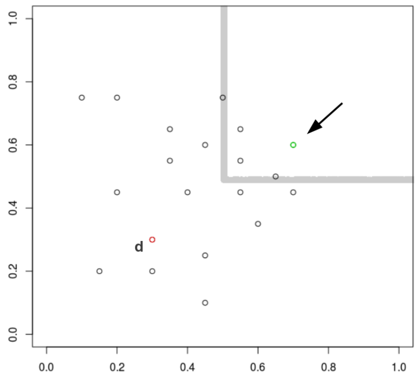


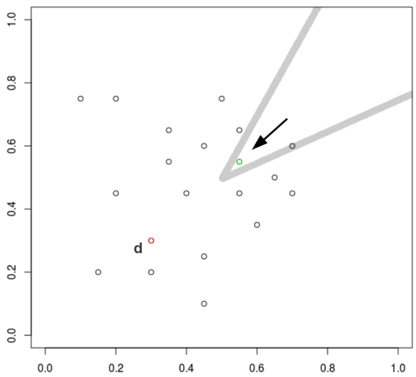
Figure 3.14: Level 1 sets for the objective function and growth directions as w_1 changes.

We illustrate the ideas through an example, see Fig. 3.15. Given a set of alternatives and the disagreement point $d = (0.3, 0.3)$, in red, we look for the solutions of the game (represented in green) when we change the cooperativeness and competitiveness parameters. The associated isovalue functions for level 1 are shown in grey. We distinguish three cases: in Fig. 3.15(a), the solution is $x^* = (0.7, 0.6)$, and this happens whenever the cooperativeness parameter is $w_1 \geq 0.5$; in Fig. 3.15(b), the solution is $x^* = (0.55, 0.55)$, and this happens whenever

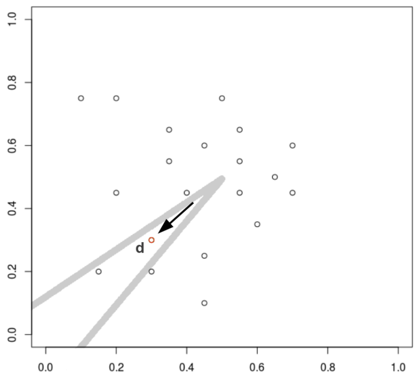
$0.3 < w_1 < 0.5$. Finally, in Fig. 3.15(c), the solution is $x^* = d = (0.3, 0.3)$, which happens whenever $w_1 \leq 0.3$.



(a)



(b)



(c)

Figure 3.15: Solution depending on w_1 and w_2.

3.6 Computational experience

In this Section we demonstrate the assessment of the solution concepts explained in Sections 3.3, 3.4 and 3.5, based on simulations.

We maintain the same assumptions from Section 2.5.2. However in this case, we simulate an environment in which a user (B_1) interacts simultaneously with two robotic agents (A_1 and A_2). Both agents make their decisions based on the ARA framework considering their opponents as non-strategic thinkers, thus they are in the first level of the ARA hierarchy. We assume that the user interacts with both agents simultaneously. The initial conditions of the environment are not relevant for this simulation and both agents start with the same battery level and environmental conditions.

3.6.1 Competitive scenarios

Through this experiment, we want to show how a level-1 agent behaves facing first another level-1 agent and, secondly, a level-0 agent. Thus, in the first case, both agents make their decisions based on the ARA framework considering their opponents as non-strategic thinkers, whereas in the second only one of them does. In both cases there is a user interacting with the incumbent agents. We expect that when all the decision agents within a society perform a level-1 ARA analysis, they perform as under a fictitious play model, see Brown (1951), which leads them, after a sufficiently long performing period, to a Nash Equilibria.

In Figs. 3.16 and 3.17, we may see the reaction of a level-0 agent and a level-1 agent (rows in both cases), respectively, facing a user (columns). We may appreciate that the level-1 agent behaves more coherently performing more often actions like *apologize*, *asking for help* or *complain*, or reducing the frequency of actions like *asking for playing*, *flatter* or *do nothing* when it is attacked.

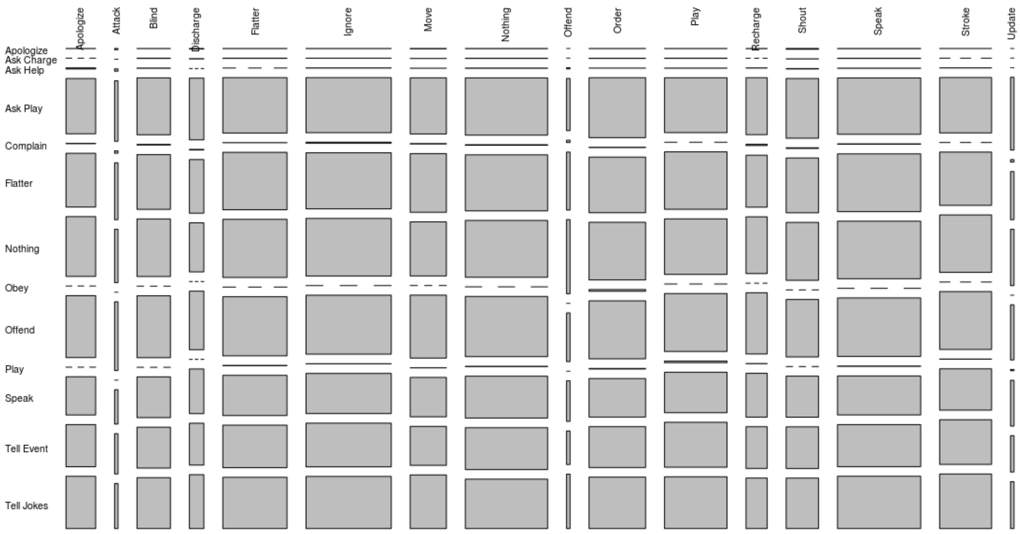


Figure 3.16: Level-0 agent reacting to user actions.

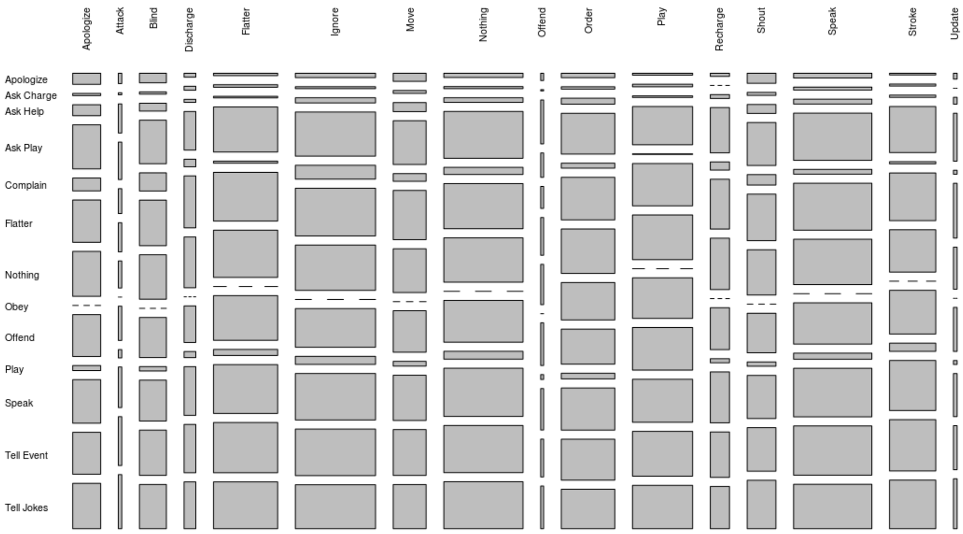


Figure 3.17: Level-1 agent reacting to user actions.

Given that the user action and the environmental states are perceived simultaneously by all agents, the utilities obtained should not be considerably different, and any existing difference would be due to the action implemented by the faced agent. Both level-1 agents achieve approximately the same utility when facing each other, as we expect them to be under a Nash Equilibria (first quartile $\approx$ 0.42, median $\approx$ 0.48, third quartile $\approx$ 0.53), whereas slightly lower facing a level-0 agent (first quartile $\approx$ 0.4, median $\approx$ 0.46, third quartile $\approx$ 0.52). The utilities obtained by the level-0 agent are slightly higher (first quartile $\approx$ 0.41, median $\approx$ 0.47, third quartile $\approx$ 0.53). Making a similar comparison, the level-1 agent who faced a sophisticated agent computed lower expected utility values (first quartile $\approx$ 0.057, median $\approx$ 0.0635, third quartile $\approx$ 0.072), than facing a level-0 agent (first quartile $\approx$ 0.06, median $\approx$ 0.068, third quartile $\approx$ 0.077). From these data we may conclude that all agents obtain approximately the same expected utility and utility levels since the same user action and environmental states are perceived simultaneously, but small differences are noted like slightly higher utilities are reached facing sophisticated peers than non-strategic ones, whereas lower levels of expected utility. This could be due to the fact that level-1 agents perform as under a Nash Equilibria obtaining sufficiently high values of utility slightly outperforming other situations, in which agents, that are not in equilibrium, expect to reach higher values.

3.6.2 Cooperative scenarios

Within this cooperative situation, whenever the user performs an action within the user's interacting group (*speak*, *ignore*, *order* and *play*), see Fig. 2.3, we would consider it as a potentially conflicting situation. The two agents will establish communication with a third robotic agent, used as CTTP. This collects all the information needed from the agents and compute the solution described in Section 3.5, which depends on the society's cooperativeness and competitiveness parameters. Each agent j has an independent value for its cooperativeness parameter w_{j1}, which, we assume that, will depend on the level of utility obtained in the previous iteration, so that: $w_{j1} = (1 - u_j(a_{t-1}, b_{t-1}, e_{t-1}))$, with $w_{j2} = 1 - w_{j1}$. In other words, if the agent is satisficing its objectives, it will contribute positively to the society's cooperativeness parameter. Note that the higher values of $u_j(a_{t-1}, b_{t-1}, e_{t-1})$, the closer it will be to 1, so that w_{j1} will be smaller. As explained above the CTTP would compute the society's competitiveness based on each agents parameters.

In this case we want to study which is the solution suggested by the CTTP and whether that solution improves, in utility and expected utility terms, the ARA-1 solution, and how the implemented action is affected by those values of the society's competitiveness. For that purpose, we have computed the cooperative and non-cooperative solutions at each time-step, while the cooperative one is the only solution applied during the simulation.

We have considered that the disagreement point will be where both agents implement the ARA-1 solution, as in Section 3.3. If there is no action with a higher expected utility than the disagreement point for one of the agents, both agents would choose the disagreement point action.

Figures within this Section corresponds to agent A_1's. Those from agent A_2 have been omitted to avoid redundance as both agents perform similarly.

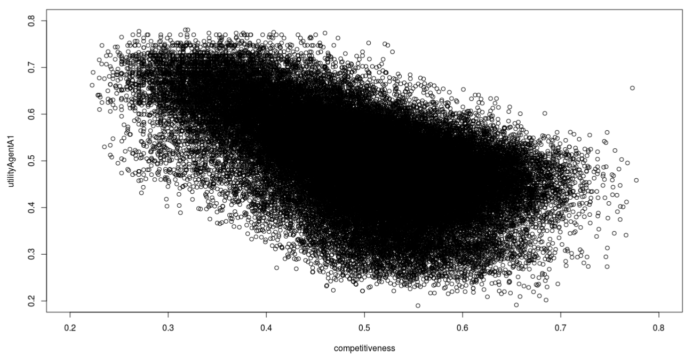


Figure 3.18: Society's competitiveness evolution in comparison with the utility for agent A_1.

Fig. 3.18 shows the evolution of the society's competitiveness parameter related with the utility of the consequences obtained by agent A_1. We may appreciate that the less competitive the society behaves, the more utility the agent experience, and vice versa. Given the assumptions made, society's competitiveness parameter w_1 obeys to:

$$w_1 = \frac{1}{2}(1 - u_1) + \frac{w_{22}}{2}.$$

In Fig. 3.19, we may appreciate that the more competitively the society behaves the expected utility that each agent obtains from the set of alternatives reaches lower levels, as the actions selected are closer to the disagreement point.

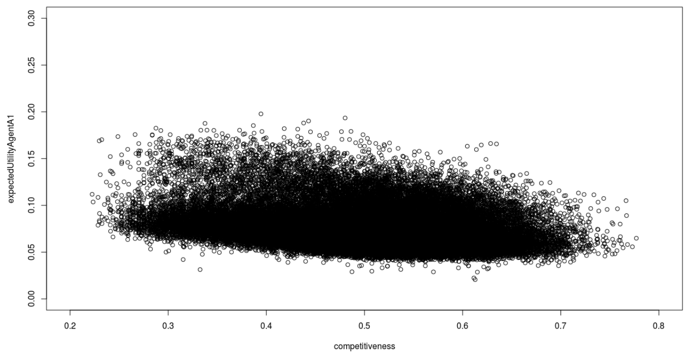


Figure 3.19: Impact of society's competitiveness on agents' expected utility.

Fig. 3.20 shows how agent A_1's actions are performed under different values of the competitiveness parameter. Actions *complain* and *ask for charge* are those perfomed with highest values of competitiveness. This is due to the fact that during the previous interaction, the utility obtained from the consequences was low enough to trigger such actions, see assumptions made in Section 2.5.2. Actions *flatter* and *speak* are mainly performed when the competitiveness parameter is slightly higher than for the rest of actions. This could be due to the fact that they have been used as a reaction, under cooperative situations, to some unsatisfying outcomes.

Fig. 3.21 describes how a cooperative agent achieves higher values of expected utility than an ARA-based agent. From statistics obtained, we may appreciate that, under the ARA framework agents compute lower levels of expected utility (first quartile ≈ 0.043, median ≈ 0.047, third quartile ≈ 0.054) than under a cooperative attitude (first quartile ≈ 0.059, median ≈ 0.069, third quartile ≈ 0.082). Based on these data, we may conclude that, despite the entailed variation

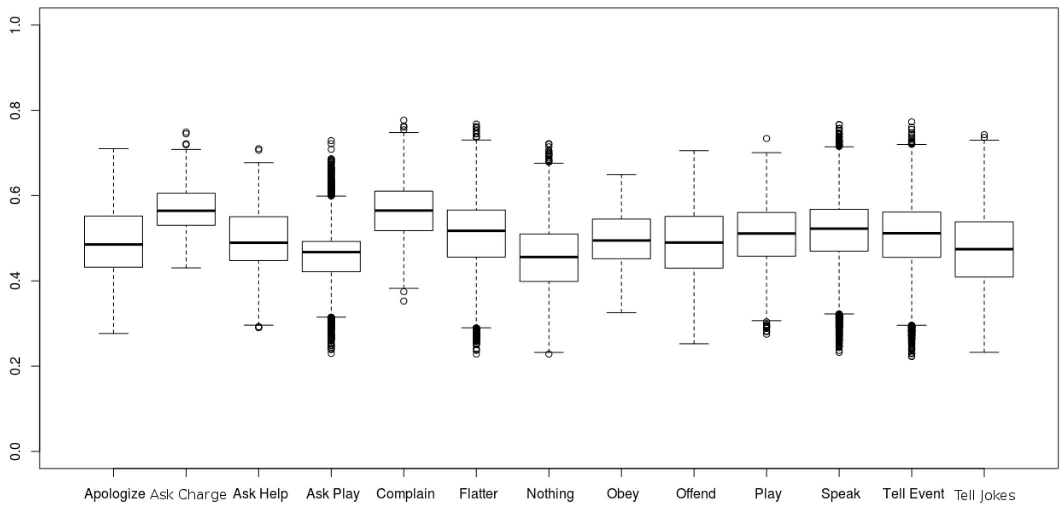


Figure 3.20: Competitiveness parameter impact on agents' chosen actions.

of the competitiveness parameter, using it within the solution described in (3.15), impacts positively on the expected utility values, outperforming those of an ARA-1 solution.

From the statistics computed, we may observe that under the ARA framework the utilities obtained from the consequences are slightly lower (first quartile $\approx$ 0.437, median $\approx$ 0.494, third quartile $\approx$ 0.554) than those obtained under a cooperative attitude, being the cooperative approach a better solution (first quartile $\approx$ 0.459, median $\approx$ 0.507, third quartile $\approx$ 0.566), in terms of utility, for a society of agents.

Finally, in Figs. 3.22 and 3.23 we may observe the different reaction of an agent (rows) within a cooperative situation and under the ARA framework, respectively, while interacting with the same user (columns). We appreciate that under cooperative situations, see Fig. 3.22, actions are not as uniformly distributed as under the ARA framework. Those actions more often implemented are those which were performed under higher levels of society's competitiveness.

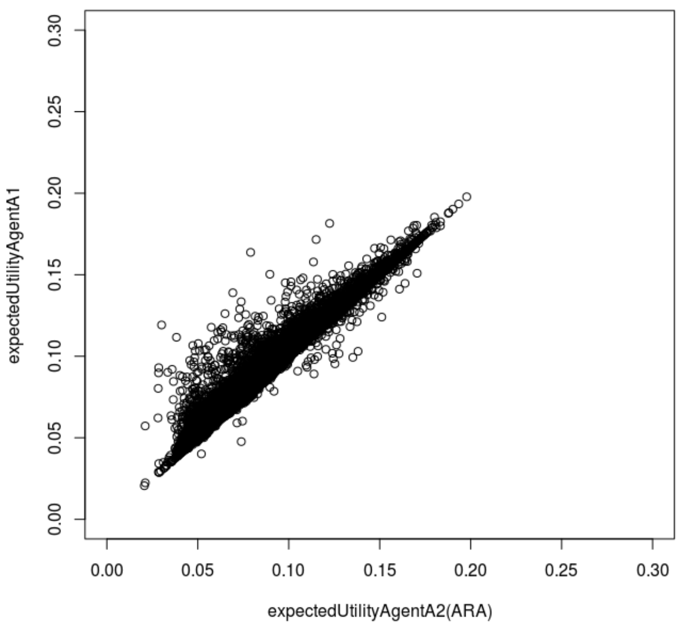


Figure 3.21: Expected utility computed by a cooperative and an ARA-based agent.

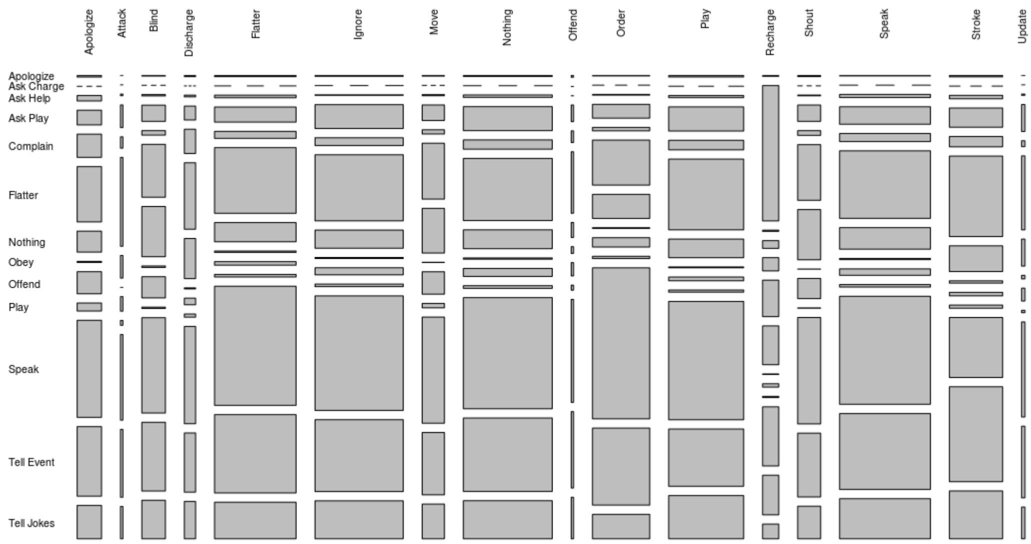


Figure 3.22: Cooperative agent's actions depending on the user's action.

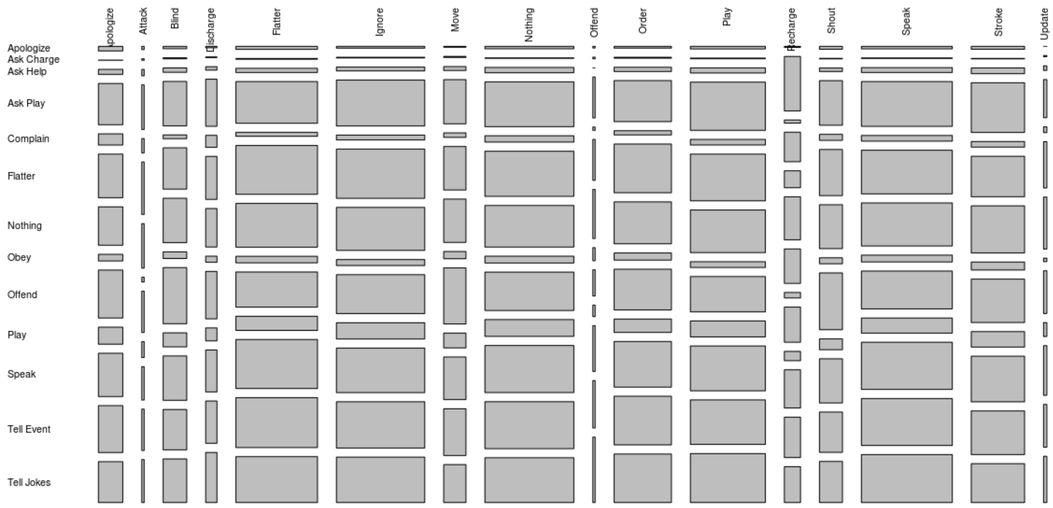


Figure 3.23: ARA based agent's actions depending on the user's action.

3.7 Discussion

Throughout this chapter we have described a behavioral model for an autonomous agent, which imperfectly processes information from its sensors, facing several adversaries using multi-attribute decision analysis at its core, complemented by forecasting models of the adversaries. We have explored the interaction among different robotic agents and users, under competitive and cooperative attitudes, depending on the social needs of our agent. Our motivation for these models was the design of societies of robotic agents that interact among them and with one or more users. Those agents could be used as interactive robotic pets, robotic babysitters and teaching assistants or cooperative caregivers for the elderly.

Regarding competitive attitudes, we described two scenarios in which a Computerised Trusted Third Party handles potential conflicts in case we opt for a game theoretical solution. But we prefer to promote the ARA solution, as it avoids the common knowledge assumption.

Within cooperative scenarios we aimed at supporting a society of agents, which might behave cooperatively towards one or several users, with the possibility of moving from a competitive attitude towards a cooperative one. We have introduced a solution concept which aims at maximizing a certain distance function from the disagreement point has been provided. The distance function used is parametrized with w_1 and w_2, which measures the degree of cooperativeness and competitiveness of our robotic agents, respectively. Based on these parameters, the agent would move from a cooperative attitude towards a competitive one, or vice versa.

We have demonstrated that the society's competitiveness parameter has an impact on the actions implemented by the agents, and that, using the cooperative solution, the expected utility and the utility of the consequences that the agents receive are higher than under the non-cooperative solution.

Chapter 4

An Affective Model for an Autonomous Decision Agent

4.1 Introduction

Throughout Chapters 2 and 3, we have described a behavioral model for an autonomous agent who makes decisions within different interactive situations exploring its social needs. Our agent made its decisions essentially based on the maximum expected utility principle.

As we are expecting robots to become part of our daily lives, we would need them to be as interactive as possible, including affective mechanisms, providing more natural and satisfying human-robot interactions, see Picard (1997). Humans, as social creatures, tend to apply social rules in their interactions with every individual, even with inanimate objects, see Nass *et al.* (1997) who suggest that humans treat computers as social actors. Future robots should be able to communicate and behave affectively with humans to become collaborative companions, see Breazeal *et al.* (2004a).

Apart from using affective phenomena to improve human-machine interaction, it has been shown that emotions may have a direct impact on decision-making processes, see e.g. Busemeyer *et al.* (2006). Advances in areas such as affective decision-making (Damasio, 1994), neuroeconomics (Glimcher *et al.*, 2008) and affective computing (Picard, 1997) are based on this principle. This chapter aims at providing a model for an autonomous agent that makes decisions partly influenced by affective factors when interacting with humans and other agents.

As it happens with most of the proposed computational emotional models in the literature, see Section 1.4.4, they require taking into account a big number of variables and complex interactions which increase the complexity of the system. We present here a model which aims at providing a believable approach implemented within non-expensive computational platforms.

Our development is based on our previous multiobjective utility model, with weights regulated by the agent's affective states, see Chapter 2. Our motivation for this work is not to provide a more accurate descriptive model, but rather to use affection as a basic element within a decision-making process that could lead us to improve interaction between our robotic agent and users, possibly making our agent more attractive, and therefore, prolonging the interaction. This would facilitate their adoption in application areas such as companion pets to provide emotional support to the elderly or autistic kids, or as educational support within an edutainment framework.

We first introduce the basic elements of our framework and the affective model in Section 4.2. In Section 4.3, we provide an implementation of such model. Section 4.4 describes a set of experiments to validate its outcomes. We end up with some discussion and open issues.

4.2 Our approach

Through the term affective model we may consider a number of phenomena, which typically include *emotions*, *mood*, *personality traits*, *attitudes* and *motivational states*, see Moshkina (2011). As argued in Russell & Barrett (1999), to date, there is no unified model for the computational generation of all these affective parameters, as authors' interests have focused on a wide variety of purposes and goals within the intelligent agents field, resulting in a large amount of approaches which are either biologically, cognitively or neurologically inspired. Most of the existing computational models are based on emotions and mood, as in van Breemen (2004), Wada & Shibata (2007) and Gratch & Marsella (2004), or on emotions, mood and additional affective variables, as in Kirby *et al.* (2010)

or Elliott (1992). One of the most complete models is FLAME, see El-Nasr *et al.* (2000), which includes emotions, mood, attitudes and motivational states. Within our work, we have considered a model that is a compound of emotions and mood, leaving for future work the addition of the remaining phenomena. Our agent's behavior will be influenced by these affective factors in different ways. We provide a brief review on how the different affective mechanisms are considered in the literature and how we have modeled them within our approach.

4.2.1 A brief review

Depending on whether emotions are considered to belong to a discrete or a continuous space, there are different approaches on how to model and express them. The term discrete or "basic" emotions has been found to be quite controversial as researchers do not agree on how to categorize them, see Russell & Barrett (1999), and the amount of emotions to be considered, ranging from two to nine, see Ekman (1999) and Ortony (2002). Some others have characterized emotions in terms of continuous dimensions, see Russell & Barrett (1999), describing them as points located in a typically two-dimensional space (valence, arousal). Based on FLAME, we considered the use of discrete emotions adopting a categorical approach. For our purposes, we find the categorical approach to emotions more suitable, as we were aiming at a small set of easily recognizable emotions, and as it allows modeling each emotion as a mechanism with its own adaptive function, antecedents, and a set of responses, see Ekman (1992), Keltner *et al.* (2003), Panksepp (2000) and Izard & Ackerman (2000).

Inspired by FLAME, we initially attempted to produce a model with fourteen discrete emotions. However, the resulting interaction was confusing, and it was difficult to grasp the conveyed emotional meaning. We decided to streamline the emotional component by limiting it to four basic emotions: hope, fear, happiness and sadness. The first two are experienced prior to making the incumbent decision, based on the forecasted expectations, whereas the other two are experienced after the decision is made, based on the decision consequences, following ideas from Loewenstein & Lerner (2003). The choice of this core set of emotion types is not arbitrary, they have been chosen considering that each couple is composed of antagonist emotions, and their adaptability to our context.

Following the *core-relational theme* concept from Lazarus (1991a), which identifies the set of events that induce an specific emotion, we assume that hope arises when “fearing the worst but yearning for better”; fear emerges when “facing an immediate, concrete, and overwhelming physical danger or threat”; happiness is usually induced when “making reasonable progress towards the realization of a goal”; while, finally, sadness occurs when “having experienced an irrevocable loss”. Contextualizing these core-relational themes within our model, still in qualitative terms, we shall consider that:

- Hope is experienced when we expect something good to happen in the future. The more sure we feel about it, the more hopeful we are.

- Fear is experienced when something bad in the future is expected. Again, the more sure we feel about it, the more fearful we become.

- We get happier when something good, evaluated in terms of its impact on the agent objectives, has just happened. We get happier, the more unexpected the consequences were.

- Finally, we get sadder when something bad has happened. The more surprising it was, the sadder we become.

We assume that our agent may not simultaneously feel opposite emotions (happy and sad, hopeful and fearful), see Brehm & Miron (2006) where theories and evidences about the mutual replacement of opposite emotions are reviewed.

Our model will also entail a concept of mood. Mood is considered to be related to, but different from, emotions, see Sloman (2002). Typically, mood is distinguished from emotion as being longer-lasting and more global, see Lazarus (1991b). As argued in Davidson (1994), mood may be seen as the agent’s affective background, and emotions as the phasic perturbations of this background activity. Mood has been shown to make impact on cognitive, perceptual and behavioral processes, see Gratch & Marsella (2004). Based on Picard (1997), agents with a bad mood tend to experience more negative emotions, and vice versa. For these reasons, mood will play an important role within our model leading the influence on the agent’s behavior selection process.

The most popular categorizations of personality traits are those developed by McCrae & Costa (1997) and Goldberg (1990), which are similar enough to be treated as a single taxonomy, see Moshkina (2011). They use five dimensions: Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism, and are known as "The Big Five" or "Five-Factor Model". Despite that personality traits are considered by Ortony (2002) as a way to distinguish people in their experience of emotions and their behavior, we have decided not to include them in our model and consider them in future work, due to the computational complexity increase they would entail.

Motivational states comprise any internal state, or physical condition, that tends to drive the agent to take actions to fulfill important needs, having an effect on emotions, see Bolles & Fanselow (1980). For example, pain may inhibit fear; hunger may inhibit happiness; and so on. Bui (2004) considers four variables to describe its agent's physical conditions: hunger, thirst, fatigue and pain, which evolve dynamically over time. The robot developed by Breazeal (2003) uses its motivational states within the processes of behavior and attention selection. We are assuming that our robotic agent does not experience independent feelings such as hunger, fatigue, etc. In contrast, its behavior selection will be based on its preference model which is influenced by the mood, and on an impulsive behavior triggered by high intensities of certain emotions, see Loewenstein & Lerner (2003). Frijda (1986) introduced the idea of *action tendencies*, describing them as "states of readiness to execute a given kind of action, which is defined by its end result aimed at or achieved". Thus, in cases of impulsive behavior, each emotion shall trigger a set of actions associated with it. In our model, the agent's behavior shall not be directly dependent on its physical conditions but rather on its mood and the intensities of its emotions.

Individuals have an affective attitude towards an important amount of things as they are associated with past experiences, as in the famous experiments on dogs where conditioned fear was demonstrated by Pavlov (1927). We shall not consider attitudes as an essential affective component, as we have already included learning from past experience within our decision-making model, see Section 2.4.2.

4.2.2 The model

Figure 4.1 provides a schematic view of the interrelations and relations over time within our affective model. The four basic emotions, as well as the *mood*, evolve with time t. We group them into *expected* ($expEm_t$) and *immediate* ($immEm_t$) emotions, as in Loewenstein & Lerner (2003). *Expected* emotions are evaluated before the decision is made. *Immediate* emotions are evaluated after such decision is made and the entailed consequences are experienced. Based on those ideas from Loewenstein & Lerner (2003), we define interdependencies among them, as shown in Fig. 4.1: $expEm_t$ depends on $expEm_{t-1}$, and $mood_{t-1}$; $immEm_t$ depends on $immEm_{t-1}$, and $mood_{t-1}$; finally, $mood_t$ depends on $expEm_{t-1}$, $immEm_{t-1}$, and $mood_{t-1}$.

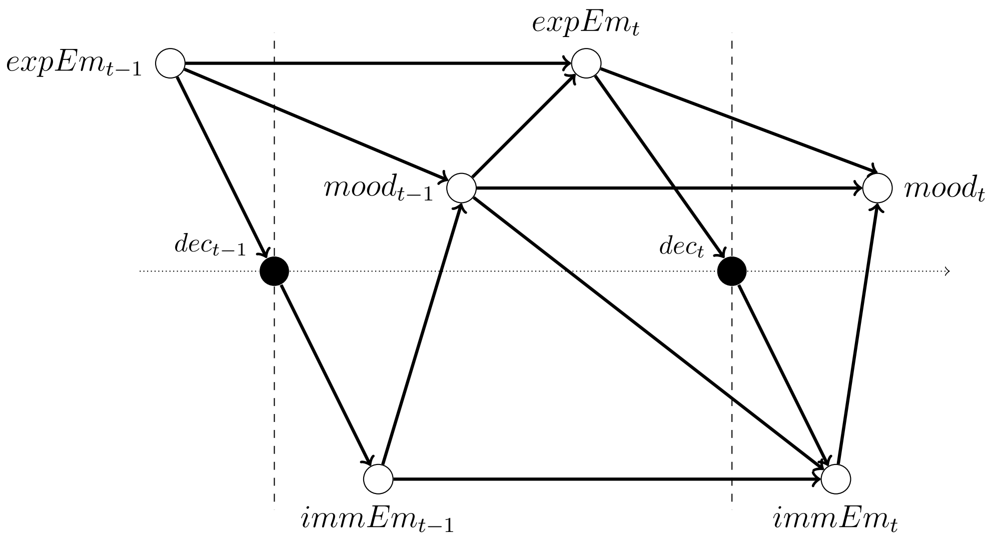


Figure 4.1: Schematic view of emotions' and mood's influence.

Following the notation in Chapter 2, our agent will choose an action $a_t \in \mathcal{A} = \{a_1, \ldots, a_m\}$ and obtain consequences depending on different states $\theta_t \in \{\theta_1, \ldots, \theta_z\}$, with $\theta_t = (b_t, e_t)$ where b_t is the inferred user action and e_t is the environmental state at time t. The agent obtains consequences $c_i(a_t, b_t, e_t)$ for $i = 1, \ldots, l$, being l the number of objectives. These consequences are evaluated through the utility function

$$u(c_1, c_2, \ldots, c_l) = \sum_{i=1}^{l} w_i u_i(c_i), \tag{4.1}$$

with $w_i \geq 0$, $\sum_{i=1}^{l} w_i = 1$. Our agent makes predictions based on a forecasting model

$$\begin{aligned} p(e_t, b_t \mid a_t, (e_{t-1}, a_{t-1}, b_{t-1}), (e_{t-2}, a_{t-2}, b_{t-2})) = \\ = p(e_t \mid b_t, e_{t-1}, e_{t-2}) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}). \end{aligned}$$

The agent maximizes the predictive expected utility, i.e. implements the alternative solving

$$\max_{a_t \in \mathcal{A}} \psi(a_t) = \int \int u(a_t, b_t, e_t) \times p(e_t \mid b_t, e_{t-1}, e_{t-2}) \times p(b_t \mid a_t, b_{t-1}, b_{t-2}) db_t, de_t.$$

The main idea in this Chapter will be that after the action of the user and the evolution of the environment are observed, the expected and immediate emotions will arise and the mood will evolve affecting the preference model's weights w_i. If either the expected or the immediate emotions exceed a certain threshold an impulsive behavior will be triggered, performing the agent the action tendency associated with the incumbent emotion. If such threshold is not reached, the agent will aim at maximizing the predictive expected utility and implementing the corresponding action. Finally, the agent will express the emotion with the highest intensity through its RGB led and servos. Through this model we plan our agent's activities according to the loop in Fig. 4.2, see Plutchik (2001) for a similar approach, which is an extension to that described in Fig. 2.1.

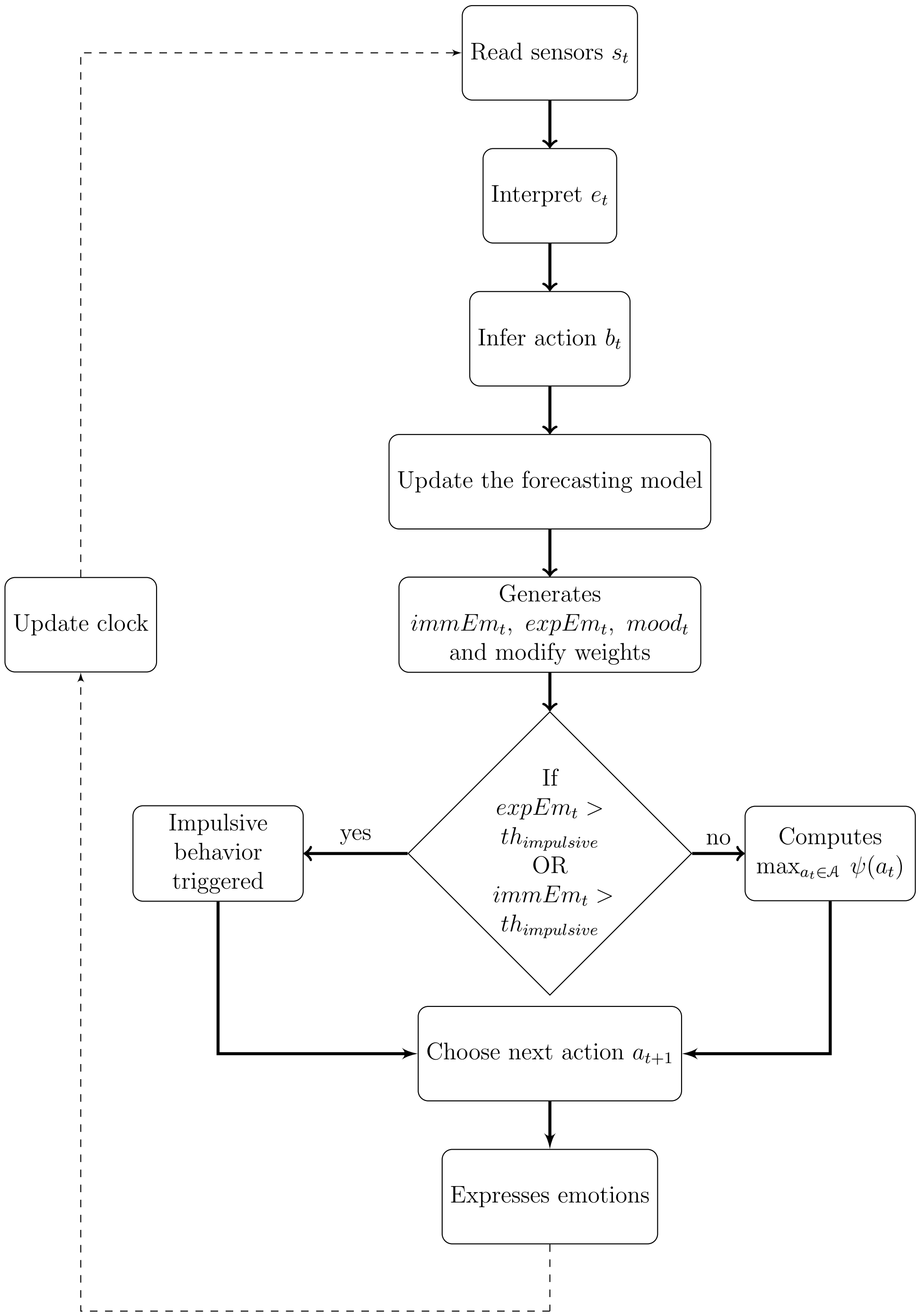


Figure 4.2: Schematic view of the behavioral model.

4.2.3 Modeling emotions

To adapt the core-relational themes described in Section 4.2 to our model, we shall consider concepts such as desirability, certainty, expectations and surprise, as we describe.

Emotions prior to making decisions

Before making the decision, our agent:

1. Computes, for each action a_t, its expected utility:

   $\psi(a_t) = \sum p(\theta_t \mid a_t) u(a_t, \theta_t)$, where $\theta_t = (b_t, e_t)$.

2. Then, it looks for the action which maximizes expected utility:

   $a_t^* = \arg\max_{a_t \in \mathcal{A}} \psi(a_t)$.

3. It obtains also the corresponding maximal expected utility:

   $\psi(a_t^*) = \psi_t^*$, which we designate the expectation.

Other relevant values that the agent will compute are:

- The most likely scenario, given the action to be implemented:

  $\theta_t^* \equiv \arg\max_{\theta_t \in \{\theta_1, \ldots, \theta_z\}} p(\theta_t \mid a_t^*)$.

- The uncertainty of the expectation, which is the utility variance for the optimal alternative:

  $var_t = \sum p(\theta_t \mid a_t^*)(u(a_t^*, \theta_t) - \psi_t^*)^2$.

As mentioned before, we call *expected* to the incumbent emotions. We assume that opposite expected emotions may not appear simultaneously: there is a threshold th_{exp} which will determine whether the emotion is positive (hope) or negative (fear). Expected emotions will be based on a function which depends on $(\psi_t^*, var_t, th_{exp}, expEm_{t-1}, u^M, u_m)$, where u^M and u_m are the maximum and minimum utilities attainable, which, without loss of generality, we calibrate to $u^M = 1$ and $u_m = 0$. $expEm_{t-1}$ would be the value of the last expected emotion experienced. Qualitatively, we expect:

- if $\psi_t^* > th_{exp}$, i.e. the agent is expecting sufficiently good consequences

$$\text{hope}_t \ = e(\psi_t^*, var_t, th_{exp}, expEm_{t-1}, 1, 0) \text{ and fear}_t \ = 0,$$

  hope_t increases with higher values of ψ_t^* and lower values of var_t. The last expected emotion $expEm_{t-1}$ is used to soften the transition between emotional states, avoiding a somewhat bipolar behavior.

- else, the expectations are low

$$\text{hope}_t \ = 0 \ \text{ and fear}_t \ = -e(\psi_t^*, var_t, th_{exp}, expEm_{t-1}, 1, 0),$$

  which grows with lower values of ψ_t^* and lower values of var_t. As before, the last expected emotion $expEm_{t-1}$ is used to soften the transition between emotional states.

The threshold th_{exp} will be influenced by the mood of our agent.

Emotions after making decisions

After making the decision, our agent evaluates the consequences and how surprising they were:

1. After having implemented a_t^*, our agent faces the evolution of the environment and the action of the user $\theta_* = (b_t^*, e_t^*)$.

2. He, then, evaluates the consequences, obtaining the utility attained, $u(a_t^*, \theta_*) = u_t^*$, which will be the desirability of the consequences.

3. Finally, it computes a surprise factor based on a distance between the observed evolution θ_* and the most likely θ_t^*, which we represent by $s_t^* = d(\theta_t^*, \theta_*)$, for a certain distance-like function specified below.

As previously mentioned, we call *immediate* to the incumbent emotions. Again, we assume that opposite immediate emotions may not hold simultaneously, so that there would be a threshold th_{imm} which will determine whether the emotion is positive (happiness) or negative (sadness). Immediate emotions will be based on a function with arguments $(u_t^*, s_t^*, th_{imm}, immEm_{t-1}, 1, 0)$, where $immEm_{t-1}$ would be the value of the last immediate emotion experienced. Qualitatively we have:

- If $u_t^* > th_{imm}$, i.e. the utility from the consequences is high enough

$$\text{happ}_t \;= i(u_t^*, s_t^*, th_{imm}, immEm_{t-1}, 1, 0) \text{ and } \text{sad}_t \;= 0.$$

  happ_t increases with higher values of u_t^* and s_t^*. The last immediate emotion simulated $immEm_{t-1}$ is used to soften the transition between emotional states.

- else, the consequences were not good enough

$$\text{happ}_t \;= 0 \;\text{ and } \text{sad}_t \;= -i(u_t^*, s_t^*, th_{imm}, immEm_{t-1}, 1, 0),$$

  which grows with lower values of u_t^* and higher values of s_t^*. The last immediate emotion $immEm_{t-1}$ is used to soften the transition between emotional states.

Again, the threshold th_{imm} will be influenced by the mood of our agent.

4.2.4 Modeling mood

Mood will be represented as a variable that increases when events and emotions, considered as positive, i.e. exceed an specific threshold described next, occur, and decreases with negative ones, see Frijda (1994) and Davidson (1994). Following the ideas from Kirby *et al.* (2010), El-Nasr *et al.* (2000) and Gratch & Marsella (2004), we consider mood as an aggregate function of the emotional states over the last r time periods. Qualitatively, we have:

$$\text{mood}_t = m\left(\sum_{i=0}^{r} \left(e_{t-i}(\cdot) + i_{t-i}(\cdot)\right)\right),$$

where r will depend on the computational limitations of the platform in which the model is implemented, and $e_{t-i}(\cdot)$ and $i_{t-i}(\cdot)$ are the functions that define the expected and immediate emotions respectively. m will increase with positive values of $e_{t-i}(\cdot)$ and $i_{t-i}(\cdot)$, and decrease otherwise.

Mood has a direct impact on emotions by lowering the activation threshold of particular emotions, see Frijda (1994), and by increasing the probability that similarly valenced emotions are triggered, see Davidson (1994). Therefore, the activation thresholds th_{imm} and th_{exp}, which, respectively, determine whether a positive or a negative emotion is activated, will be influenced by the agent's mood.

4.2.5 Behavior selection

As described in (4.1), our agent's preferences will be regulated by a weighted additive utility function, with objectives ordered through $w_1 \geq w_2 \geq w_3 \geq w_4 \geq w_5$. We assume that the agent will devote more resources to the upper objective as it gets into a better mood (more hopeful, happier) and will focus on lower objectives as it gets into a worse mood (more fearful, sadder), see Bower (1991) and Wright & Bower (1992). To attain a smoother behavior, we shall assume that current weights have an influence over weights in the next period, through

$$w_t^i = g_i(w_{t-1}^i, mood_t),$$

for each objective $i \in \{1,\ 2,\ 3,\ 4,\ 5\}$. The function applied to the first objective will be the opposite to that applied to the fifth objective as the increment (or decrement) of one of those objective implies the decrement (or increment) of the other, and similarly, between the second and fourth objectives. The third objective will remain unvaried.

Loewenstein & Lerner (2003) describe that low and medium intensities of emotions have some kind of advisory role in the decision-making process of an agent, while high intensity emotions lead to a more impulsive behavior. Such "out of control" behavior would be based on an action tendencies set, which specifies that each intense emotion experienced shall trigger a set of actions associated with it, see Frijda (1986).

4.3 Implementation

We provide now parametric forms for the functions proposed above. We have based our implementation qualitatively on exponential smoothing, see Brown (1963). This modeling technique allow us to deal with time series, where $\{x_t\}$ is the raw data sequence, s_t represents the output at time t, and α the smoothing factor.

$$\begin{aligned} s_0 &= x_0, \\ s_t &= \alpha \times x_t + (1-\alpha)s_{t-1}. \end{aligned}$$

Exponential smoothing allows us to control the growth rate of emotions and mood, smoothing their evolution over time. We have actually adopted the expressions

$$\begin{aligned} expEm_t &= \alpha \times e(\psi_t^*, var_t, th_{exp_t}, u^M, u_m) + (1-\alpha) \times expEm_{t-1}, \\ immEm_t &= \alpha \times i(u_t^*, s_t^*, th_{imm_t}, u^M, u_m) + (1-\alpha) \times immEm_{t-1}, \\ mood_t &= \gamma \times m\left(\sum_{i=0}^{r} (e_{t-i}(\cdot) + i_{t-i}(\cdot))\right) + (1-\gamma) \times mood_{t-1}, \end{aligned}$$

where α and γ are the smoothing factors, and $e(\cdot)$, $i(\cdot)$ and $m(\cdot)$ are the functions which will define the value of expected and immediate emotions and the mood, respectively, as we specify below.

4.3.1 Emotions

We consider as positive expected outcomes those whose expectation exceeds a certain threshold th_{exp_t}. For the expected emotions, we shall assume that

- if $\psi_t^* > th_{exp_t}$

$$\text{hope}_t = \alpha \times \left(\frac{\psi_t^* - th_{exp_t}}{1 - th_{exp_t}}(1 - var_t)\right) + (1-\alpha) \times expEm_{t-1}, \quad \text{and}$$
$$\text{fear}_t = 0.$$

Qualitatively, we want to capture the positive expectation of an event and the uncertainty about that event happening. To do so, we have divided $e(\psi_t^*, var_t, th_{exp_t}, u^M, u_m)$ into two terms. The first one: $0 \leq \left(\frac{\psi_t^* - th_{exp_t}}{1 - th_{exp_t}}\right) \leq 1$ normalizes the expectation obtained, measuring how good the incoming future is expected to be. The second term $0 \leq (1 - var_t) \leq 1$ indicates that the more sure the agent is about such future, the more hopeful it is. As $-1 \leq expEm_{t-1} \leq 1$, it may influence hope_t to end up being negative. In that case, fear_t will adopt hope_t value and hope_t will value 0.

- else, similarly,

$$\text{hope}_t = 0, \quad \text{and}$$
$$\text{fear}_t = -\left[\alpha \times \left(\frac{th_{exp_t} - \psi_t^*}{th_{exp_t}}(1 - var_t)\right) + (1 - \alpha) \times expEm_{t-1}\right].$$

A similar idea is applied to immediate emotions, considering as positive consequences those whose utility exceeds the threshold th_{imm_t}. Then, we have:

- if $u_t^* > th_{imm_t}$

$$\text{happ}_t = \alpha \times \left(\frac{u_t^* - th_{imm_t}}{1 - th_{imm_t}} \times s_t^*\right) + (1 - \alpha) \times immEm_{t-1}, \quad \text{and}$$
$$\text{sad}_t = 0.$$

In qualitative terms, we consider that once the agent has observed the consequences, it will be happier the better and more surprising they are. Specifically, we have divided $i(u_t^*, s_t^*, th_{imm_t}, u^M, u_m)$ into: the first term $0 \leq \left(\frac{u_t^* - th_{imm_t}}{1 - th_{imm_t}}\right) \leq 1$, which normalizes the desirability obtained from the consequences, measuring how good they were towards the agent's objectives; and the term $0 \leq s_t^* \leq 1$, which measures the degree of surprise produced by such consequences. The more surprinsing they were, the happier the agent is about them. In this case, $-1 \leq immEm_{t-1} \leq 1$ could have a value below zero, resulting in a negative value for happ_t. In that case, sad_t will adopt that value and happ_t will be 0.

- else, similarly,

$$\text{happ}_t = 0, \quad \text{and}$$
$$\text{sad}_t = -\left[\alpha \times \left(\frac{th_{imm_t} - u_t^*}{th_{imm_t}} \times s_t^*\right) + (1-\alpha) \times immEm_{t-1}\right].$$

4.3.2 Mood

Qualitatively, we assume that mood is more stable than emotions, see Lazarus (1991b). As explained, we model our agent's mood as an aggregation of emotional states over the last r periods, which we fix to $r = 2$, to preserve a certain emotional memory but being reactive to incoming events. Thus, we shall use

$$\text{mood}_t = \gamma \times \left(\frac{\sum_{i=0}^{2}(emotional\ balance)_{t-i}}{2(r+1)}\right) + (1-\gamma) \times mood_{t-1},$$

where γ is the exponential smoothing factor and $-1 \leq (emotional\ balance)_t \leq 1$ is the sum of hope, fear, happiness and sadness at time t, divided by $2(r+1)$ as we deal with two simultaneous emotions (expected and immediate) and we want $mood_t$ to be normalized between $[-1, 1]$.

As we are considering the smoothing factor as a decay process, we should take into account that mood decays slowler than emotions, see Beedie *et al.* (2005). Thus, typically, we shall set $\gamma < \alpha$. Both of them should be greater than 0.5 to provide more importance to the current state than the past one. We have empirically fixed γ as 0.6 and α as 0.8.

When an agent is in a positive mood, it will tend to be more optimistic. Thus, it will naturally expect desirable events to occur, see Fehr-Duda *et al.* (2011), lowering the threshold for arousing positive emotions, see Frijda (1994) and Velásquez (1998), and vice versa. As we have grouped emotions according to expected and immediate emotions, we shall have two different thresholds to determine the activation of each of them as they depend on different stimuli, ψ_t^* and u_t^*, respectively. Each of the thresholds increase with high values of their corresponding stimuli and negative mood, and decrease with positive mood. Note that as they are activation thresholds, their values are limited to a range within

0.25 and 0.75 to avoid extreme situations.

$$th_{exp_t} = \max\left\{0.25,\ \min\left\{0.75,\ \frac{\psi_t^* + \psi_{t-1}^* + \psi_{t-2}^*}{3} - \frac{mood_{t-1}}{10}\right\}\right\},$$

$$th_{imm_t} = \max\left\{0.25,\ \min\left\{0.75,\ \frac{u_t^* + u_{t-1}^* + u_{t-2}^*}{3} - \frac{mood_{t-1}}{10}\right\}\right\}.$$

4.3.3 Surprise

As introduced in Section 4.2.3, the surprise factor s_t^* is based on some distance between the observed state, θ_*, and the most likely state, θ_t^*, given the implemented action. The Kullback & Leibler (1951) divergence has been used to approximate distances between prior and posterior distributions as a measure of surprise, see e.g. Itti & Baldi (2009) and Baldi & Itti (2010), who use the following expression to measure the surprise caused by an event D, given a model space $\mathcal{M}$ of models M:

$$S(D, \mathcal{M}) = KL(P(M \mid D), P(M)) = \int_{\mathcal{M}} P(M \mid D) \log \frac{P(M \mid D)}{P(M)} dM, \tag{4.2}$$

where $P(M)_{M\in\mathcal{M}}$ is the prior distribution over the hypothesis M; D is the observation and $P(M \mid D)_{M\in\mathcal{M}}$ is the posterior distribution given D.

The implementation we have developed is based on (4.2). The model space $\mathcal{M} = \{M_1, M_2\}$ includes:

- M_1 describes the probability assignment over the set Θ given the implemented action a_t^*, that is, $\{p(\theta_i \mid a_t^*)\}_{\theta_i\in\Theta}$, where $\theta_i = (e_t, b_t)$. See Section 2.4.2 for details on the forecasting models.

- M_2 describes the degenerate distribution at the observed θ_*, i.e.,

$$p(\theta_i) = \begin{cases} 1, & \text{if } \theta_i = \theta_*, \\ 0, & \text{otherwise.} \end{cases}$$

If $D = \{\theta_*\}$ is the observed state, we then have, if $p(M_1) = p(M_2) = 0.5$, that

$$p(M_1 \mid \theta_*) = \frac{P(\theta_* \mid M_1)}{P(\theta_* \mid M_1) + P(\theta_* \mid M_2)},$$
$$p(M_2 \mid \theta_*) = 1 - p(M_1 \mid \theta_*),$$

and consequently,

$$S(\theta_*, \mathcal{M}) = KL(P(M \mid \theta_*), P(M)) = \sum_{\mathcal{M}} P(M \mid \theta_*) \log \frac{P(M \mid \theta_*)}{P(M)} =$$
$$= \frac{P(\theta_* \mid M_1)}{P(\theta_* \mid M_1) + P(\theta_* \mid M_2)} \log \frac{\frac{P(\theta_* \mid M_1)}{P(\theta_* \mid M_1) + P(\theta_* \mid M_2)}}{0.5} +$$
$$+ \left(1 - \frac{P(\theta_* \mid M_1)}{P(\theta_* \mid M_1) + P(\theta_* \mid M_2)}\right) \log \frac{\left(1 - \frac{P(\theta_* \mid M_1)}{P(\theta_* \mid M_1) + P(\theta_* \mid M_2)}\right)}{0.5}.$$

4.3.4 Behavior selection

Affective factors are considered to bias people in their decision-making process. As mood is considered to last longer than emotions, and it has been proved that may modify people behavior towards optimistic or pessimistic judgements and choices, see Bower (1991) and Wright & Bower (1992), we shall assume that weights associated with each of the agent's objectives will vary depending on the agent's mood. Thus, we shall consider that the agent will be willing to be more self-concerned as it gets into a worse mood (more fearful, sadder), and more cooperative when it gets into a better mood (more hopeful, happier). We assume that current weights have an influence over weights in the next period, to attain a smoother behavior. All in all, we shall use

$$w_t^i = \max\{w_t^{i+1}, \min\{w_t^{i-1}, w_{t-1}^i - a \times mood_t\}\},$$

for each objective $i \in \{1,\ 2,\ 3,\ 4,\ 5\}$. Each of the weights vary with the agent's mood and a component a which measures how the mood impacts on each of them. As we want to keep the hierarchy $w_1 \geq w_2 \geq w_3 \geq w_4 \geq w_5$ among the objectives, we limit the value of each of them to be in between the previous weight and the next one, with the particular cases in which $i = 1$ or $i = 5$, we assume that $w_t^0 = 1$ and $w_t^6 = 0$, respectively. The value of a will depend on the corresponding objective.

$$a = \begin{cases} x, \text{ if } i = 1 \\ x/2 \text{ if } i = 2 \\ 0 \text{ if } i = 3 \\ -x/2 \text{ if } i = 4 \\ -x \text{ if } i = 5, \end{cases}$$

being x a value empirically fixed at 1. To normalize the expression, we use

$$w_t^i{*} = \frac{w_t^i}{\sum_i w_t^i} \quad .$$

Emotions may also impact on the behavior selection process of our agent. As described in Loewenstein & Lerner (2003), high intensity emotions arise a more impulsive behavior which may be based on the previously introduced action tendencies set, see Frijda (1986) which specifies that each intense emotion experienced shall trigger a set of actions associated with it. Thus, if at any particular time step t, the intensity of any of the emotions reaches a specific threshold ($th_{impulsive_+} = 0.9$ referring to positive emotions, and $th_{impulsive_-} = -0.9$ to negative ones), there is a set of actions associated with each of the emotions that would be triggered.

Based on Lazarus (1991a), we may determine the following action tendencies set with respect to our four basic emotions.

- **hope**: "moving towards a desired outcome". In our case, this could be translated into acting immediately to reach higher levels within the objectives hierarchy. Thus, the agent will perform hard-wired actions that ensure satisfaction like *ask for playing* or *ask for charging*, if our agent needs to be recharged.

- **fear**: "avoidance or escape". In this case, we have a very specific action that fit with the need of escaping of the agent, like *ask for shutting down* action.

- **happiness**: “share positive outcomes”. Contextualized on our agent’s environment, our agent will aim at performing *flatter* action.

- **sadness**: “inaction or withdrawal into oneself”. According to Lazarus, the action tendencies of fear and sadness are quite similar. However in the first case, it requires the agent to be more active asking the user to shut it down. For that reason, we assume that the agent will perform the *do nothing* action.

4.3.5 Emotion expression

Affective expression is one of the primary ways in which individuals communicate to others their feelings, being considered as of great importance in building and maintaining social relationships, see Ekman (2004). Through his pioneer publication in 1872, “The expression of the emotions in man and animals”, Darwin reinforced the hypothesis that there exists a relationship between facial activity and emotional state, see Darwin (1782).

More recent research has demonstrated that through facial expressions, we are able to express emotions and mood, conveying also information about the personality of participants. Most of the current work on recognizing or expressing emotions is based on the Facial Action Coding System (FACS) which provides mappings between specific muscles and emotions, see Ekman & Friesen (1977) who point out that expressions depend on the intensity of each emotion.

Many works express emotions within robotic platforms. A basic and non-expensive platform is Vikia, see Bruce *et al.* (2002), based on Delsarte’s code of facial expressions, see Stebbins (1886), where the author suggested which postures and gestures actors should perform to represent different emotional states. Kismet is a sphisticated head with active stereo vision, eyebrows, ears, eyeballs, eyelids, a mouth with two lips and a neck, see Breazeal & Scassellati (1999), which uses dimensions such as arousal, valence, and stance.

We have based our approach on that in Felix, see Cañamero & Fredslund (2001), where each of its emotional states has an associated distinctive prototypical facial expression, based on FACS, contextualized to the hardware limitations that Felix presents. We have implemented our model within the AiSoy1 robotic platform, see Section 1.1, which counts with three mini servos for the rotation of the neck, eyelids and eyebrows, a 70 miniled matrix integrated into the mouth area, as well as a RGB led in the chest area, see Fig. 4.3.

From Ekman & Friesen (1975), we have adopted the following prototypical facial expressions to be displayed:

Sadness: straight eyebrows, lowered eyelids and closed mouth slightly curved downwards.

Fear: lowered eyebrows, raised eyelids and moderately open wide mouth.

Happiness: straight eyebrows, raised eyelids and mouth curved slightly upwards.

Hope: raised eyebrows and eyelids, and open and slightly curved upwards mouth.

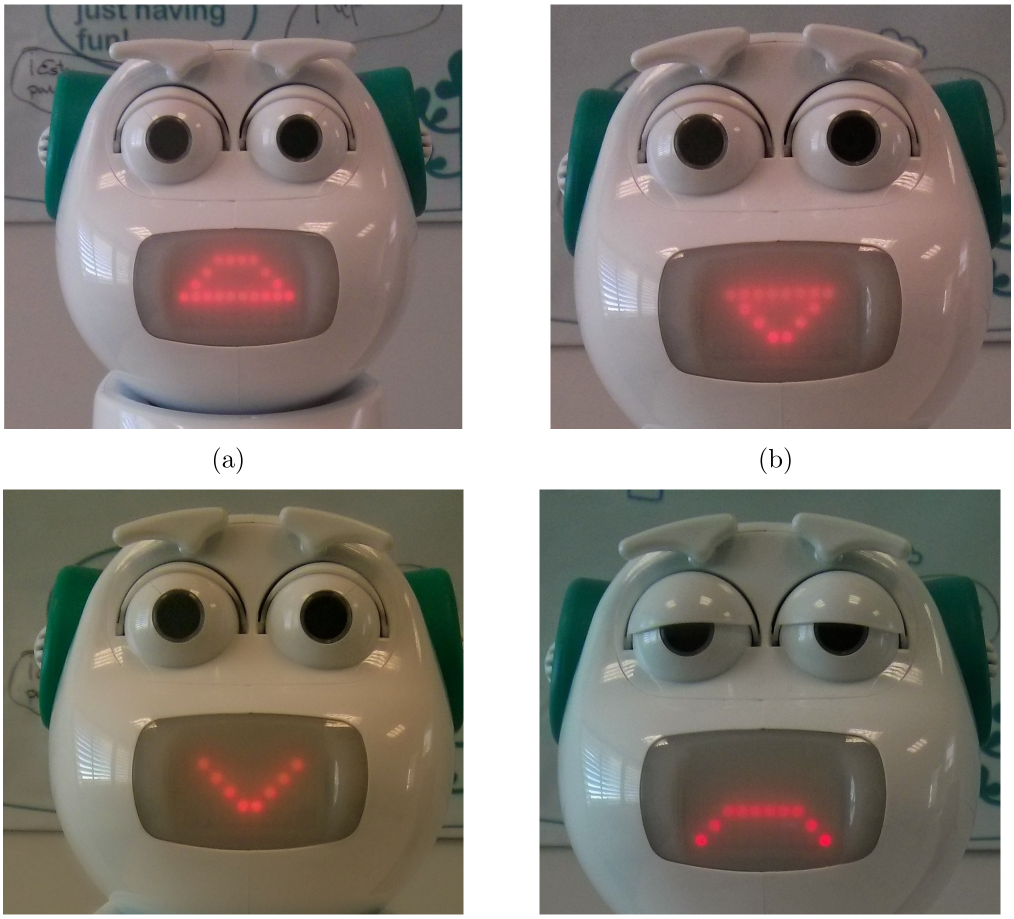
Apart from the facial expressions, our implementation uses colors as well. As described above, the AiSoy1 counts with a RGB led, so that based on Naz & Epps (2004), we shall associate sadness with blue, happiness with yellow, hope with green and fear with black. As shown in Fig. 4.2, when an action is chosen to be implemented, the emotion with highest intensity is expressed. The emotions will be expressed following the rules described above. However, in those cases in which the emotion to be expressed exceeds the corresponding impulsive behavior threshold ($th_{impulsive_+} = 0.9$ for positive emotions, and $th_{impulsive_-} = -0.9$ for negative ones), the expression to be displayed will obey the following rules:

Sadness: lowered eyebrows, very lowered eyelids and closed mouth curved significantly downwards. Lowering its head.

Fear: very lowered eyebrows, raised eyelids and wide open mouth. Shaking its head from side to side.

Happiness: straight eyebrows, raised eyelids and mouth curved significantly upwards. Rising its head.

Hope: highly raised eyebrows and eyelids, and open and curved significantly upwards mouth. Moving its head smoothly from side to side.



(a) (b) (c) (d)

Figure 4.3: (a) expresses fear, (b) hope, (c) shows happiness and (d) sadness.

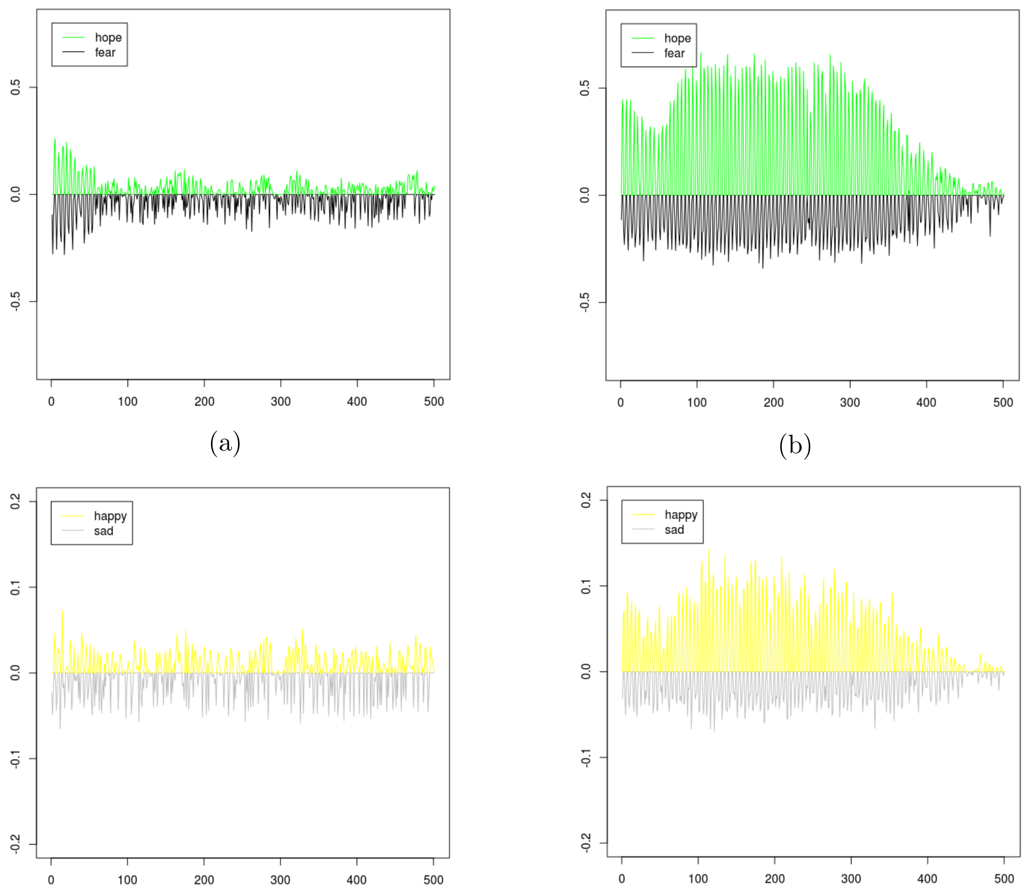
4.4 Computational experience

In this Section, we provide some simulations to assess the validity of our affective approach comparing its performance to that of an emotionless agent, see Chapter 2, when facing benevolent and malevolent humans, making the same assumptions as in Section 2.5.2. We aim at observing how the affective model impacts on the decisions of our agent. We expect gathering similar output but some variations due to dynamic changes on the agent's preference weights.

Fig. 4.4 describes the evolution of the emotions facing malevolent and benevolent adversaries, see Figs. 4.4(a) and 4.4(c), and Figs. 4.4(b) and 4.4(d), respectively. We may observe that, within the iterations considered, higher values of hope and happy are reached facing a benevolent user, and higher values of fear and sad, in comparison with those of hope and sadness, are reached facing malevolent users.

Fig. 4.5 describes higher values of mood when the adversary is behaving nicely, and lower values when she is not.

Fig. 4.6 describes the evolution of the preference model's weights depending on the faced adversary, see Figs. 4.6(a) and 4.6(b), for a bad and a good behaved user respectively. We may observe that facing a malevolent user, see Fig. 4.6(a), maintains the weights of upper objectives very low. Within the 1000 iterations shown in such figure, there are only two phases in which our agent feels comfortable enough to assign higher values to those objectives related with social interactions. However, facing a benevolent user, see Fig. 4.6(b), our agent seems to feel more secure and comfortable, therefore, it assigned resources to upper objectives more often.



(a) (b) (c) (d)

Figure 4.4: (a) and (c) refer to malevolent behavior, whereas (b) and (d) to a benevolent one.

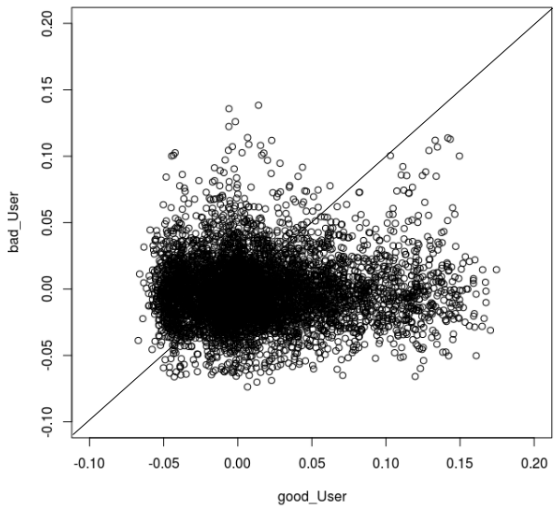
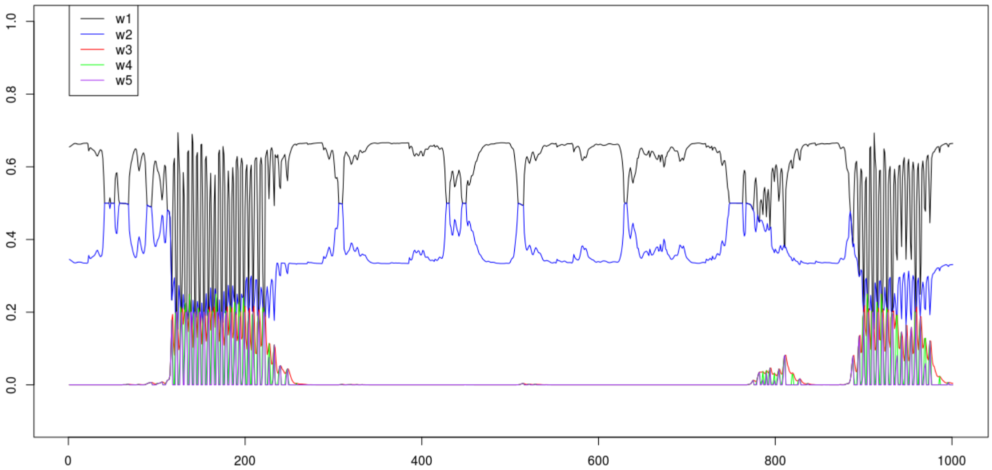


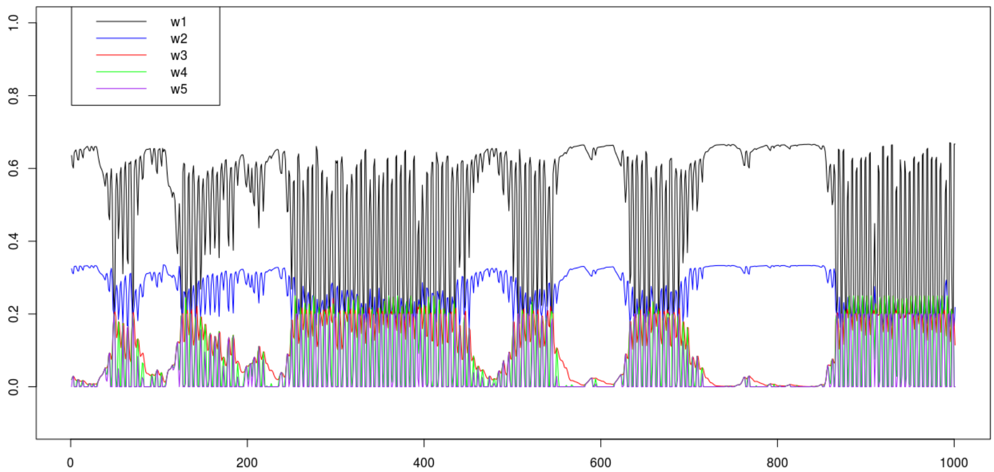
Figure 4.5: Mood generated while interacting with malevolent and benevolent users.

Then, supported by Figs. 4.4, 4.5 and 4.6, we may conclude that our affective model is coherent with different behaviors of the adversary it is facing. Following figures are related with comparing the performance of the affective agent with an emotionless one from Chapter 2.

In Table 4.1 we observe the utilities obtained from the consequences and the expected utility computed by an affective agent and an emotionless agent, while facing benevolent and malevolent users. It is shown that both agents get always less utility and expected utility when they face a malevolent user than when the user behaves nicely. It is also noticed that the affective agent gets higher levels of utility and expected utility facing a malevolent user, but not facing a benevolent one, where both agent obtain very similar outcomes. This difference could be due to the fact that preference model's weights are dynamically changing over time as we see in Fig. 4.6. So that, we may conclude that an affective agent shows more versatility facing different behaved adversaries than an emotionless agent, adapting its behavior to reach better outcomes.



(a)



(b)

Figure 4.6: Variations in preference model weights while facing a malevolent (a) and a benevolent (b) user.

			Malevolent User	Benevolent User
Affective Agent	Utilities	1^{st} *quartile*	0.38	0.444
		median	0.454	0.496
		3^{rd} *quartile*	0.52	0.578
	Expected Utilities	1^{st} *quartile*	0.4	0.429
		median	0.447	0.479
		3^{rd} *quartile*	0.511	0.557
Emotionless Agent	Utilities	1^{st} *quartile*	0.328	0.446
		median	0.382	0.507
		3^{rd} *quartile*	0.427	0.593
	Expected Utilities	1^{st} *quartile*	0.355	0.432
		median	0.383	0.485
		3^{rd} *quartile*	0.414	0.576

Table 4.1: Agent comparison facing a malevolent and a benevolent user.

In Fig. 4.7 we show the utilities obtained by an affective and an emotionless agent facing malevolent and benevolent users. We may observe that the utility obtained by an emotionless agent behaves more uniformly than that from the affective agent. Both agents achieve approximately the same levels of utility, as we may see in Table 4.1.

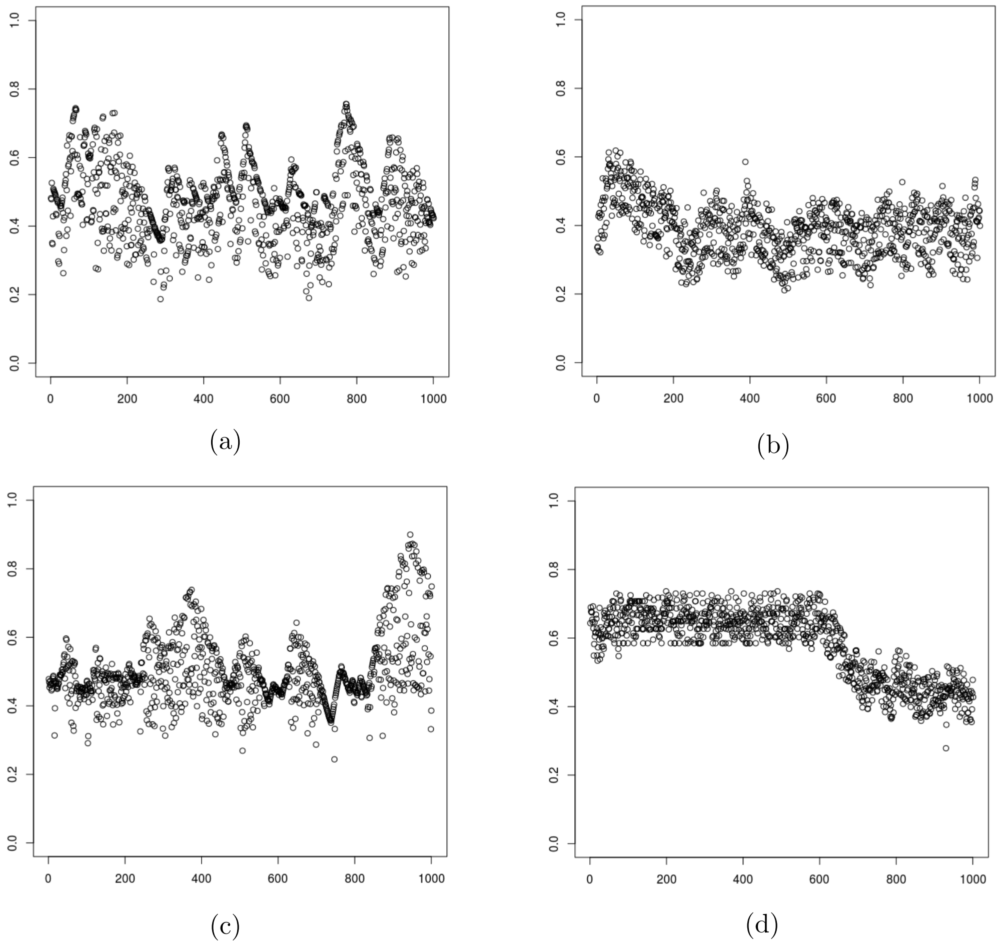


Figure 4.7: Utilites obtained by affective and emotionless agents facing bad (a) and (b) and good behaved users (c) and (d), respectively.

4.5 Implementation with AiSoy1

We have implemented this model within an AiSoy1 robot as described in Section 2.6. Within this Section we detail the functionality required to include this affective model and express the incumbent emotion at each time step, following the loop in Fig. 4.2.

Complementing those defined in Section 2.6, we describe the SDK-API and ROS nodes used to implement this model:

- *aisoy_sdk_common* and *aisoy_common* are extended including those variables related with the affective model and initializing them, like emotion intensities or thresholds variables among others.

- *aisoy_sdk_actuator* is used to access directly to an actuator and command it a movement. As an example, imagine we need to move the robot's neck to the left and then to the right, since from time to time, the robot checks the environment moving its head. To do so, AiROS would use the command *aisoy_sdk_actuator(neck, -1, 0.1)* and then *aisoy_sdk_actuator(neck, 1, 0.1)*. As parameters we include the servo to use, the position we want to move to (-1 for left, 0 centered and 1 for right) and the speed of the movement with a value between 0 and 1.

- *aisoy_sdk_performance* and *aisoy_performance* contain a set of behaviors that allow the robot to express different emotions. As an example, suppose that the emotion to be expressed is *sadness* at a high intensity, see Section 4.3.5. To do so, AiROS will send the command *aisoy_sdk_performance (sadness)* which will call the corresponding ROS node *aisoy_performance* with sadness as parameter. Within a script, it will send a set of *aisoy_sdk_actuator* commands which will make the robot to lower its eyebrows and eyelids, close its mouth curving it downwards, use its neck servos to lower its head and color purple its RGB led.

4.6 Discussion

Affective mechanisms are essential to improve the interaction among human and robotic agents, see Breazeal *et al.* (2004a), and are proved to make an impact on the decision-making process of agents, see Busemeyer *et al.* (2006). Thus, through this chapter an affective model for a non-expensive autonomous agent has been described based on emotions and mood. Rather than presenting a new and more sophisticated descriptive model, this model aims at providing an autonomous agent the ability of making decisions partly influenced by affective factors when interacting with humans and other agents, in a sense that it can possibly make our agent more believable and attractive,

Following the literature we streamlined our approach to four basic emotions, mood's influence and the possibility of triggering impulsive behavior when high intensities of certain emotions were reached.

Through a set of experiments we have shown how the evolution of emotions and mood generated changes depending on the behavior of the user the agent is facing. We have also seen how the preference model's weights vary depending on such behavior. The performance of affective and emotionless agents have been compared. Both agents obtain similar levels of utility and expected utility facing benevolent users, whereas facing a malevolent one, the affective agent reaches higher values possibly because of the dynamic variation on preference models weights.

Chapter 5

Conclusion and Open Issues

5.1 Summary

Through this thesis we have explored how the Adversarial Risk Analysis (ARA) framework may be used to support an affective decision agent facing several intelligent adversaries.

Our motivation is the design of societies of non-expensive robotic agents that interact among them and with one or more users. Those agents would include affective factors which would possibly lead them to improve interaction, making them more attractive for users, facilitating their adoption as companion pets to, e.g. provide emotional support to the elderly or autistic children, or as educational support within an edutainment framework.

We started explaining the increasing interest in designing robots to help us in our daily lives. We have reviewed how robotic agents may interact among them, outlining the key modeling approaches employed in the literature. We also reviewed social and emotional factors as we want to design believable agents.

After that, we designed an ARA based behavioral model of an autonomous agent, which imperfectly processes information from its sensors, facing an intelligent adversary (the user). This agent uses multi-attribute decision analysis at its core, complemented by forecasting models of the adversaries. That model was implemented within the AiSoy1 robotic platform. Supported by a set of simulations, we demonstrated that our model allows the agent to modify its behavior while facing malevolent and benevolent users under different objective weights, acting in a natural way. Two main goals were reached during this Chapter: designing a robust decision-making model to be implemented in non-expensive robotic platforms, and explaining how ARA may be extended to the robotic field.

Our next step in Chapter 3 was to explore the interaction among different robotic agents and users, within competitive and cooperative scenarios. Under competitive attitudes, we described two situations in which a Computerised Trusted Third Party may handle potential conflicts in case we opt for a game theoretic solution. As we wanted to avoid common knowledge assumptions, we preferred to promote an ARA solution. Within cooperative scenarios we supported a society of agents, which, depending on their past experience, could behave competitively or cooperatively towards one or several users. A solution concept aiming at maximizing a distance from the disagreement point was introduced. Such distance function is parametrized with measures of the degree of cooperativeness of our robotic agent. These parameters would lead the agent to move from a cooperative attitude towards a competitive one, or vice versa. Based on a set of simulated experiments, we have demonstrated that agents' actions are influenced by the society's cooperativeness parameter. We showed that when using the cooperative solution, the utility of the consequences that the agents receive are higher than under the ARA framework solution. Through this Chapter we were capable of solving a main goal: provinding autonomy to our agent within a society.

Finally, we considered the design of an affective model to complement our previous models, providing our autonomous agent the ability of making decisions partly influenced by affective factors when interacting with humans and other agents, see Chapter 4. It was based on our previous multiobjective utility models. We streamlined our approach to four basic emotions, mood's influence and the possibility of triggering impulsive behavior when high intensities of certain emotions were reached. We performed several simulations where our agent was facing differently behaved users and its performance was compared to that of an emotionless agent. We observe that the evolution of the emotions and the mood differs depending on the behavior of our agent's opponent; and that an affective agent shows more versatility facing different behaved adversaries than an emotionless agent, adapting its behavior to reach better outcomes. Within this Chapter, we achieved the goal of providing an autonomous agent with the ability of behaving affectively towards an user.

5.2 Open issues

In this Section we introduce some challenges that we have found during the development of this work and that, due to time limitation, we could not face within this thesis. We would approach them as short or mid-term goals.

Within our decision models we have considered a fixed finite set of user actions based on the expected interaction with an agent. Users may act differently as we predicted, so that dealing with the possibility of learning about new users' actions, based on repeated readings, and, consequently, augmenting the set $\mathcal{B}$ is one of those challenges.

The ARA models proposed within this work correspond to level-1 in the thinking hierarchy, in which the supported agent optimizes its decisions based on previous performances of the surrounding participants, considering them as non-strategic thinkers. We could explore higher level thinking ideas making our supported agent capable of facing more sophisticated adversaries. Some simulations to test how these sophisticated ARA agents would perform in realistic situations would be developed.

So far, when interacting with several users, our agent is capable of identifying which is the faced user taking a picture with its camera and using OpenCV algorithms to recognize her. In order to improve the accuracy of the indentification method, we could consider including Bayesian voice recognition algorithms, see Beigi (2011), to combine it with the current vision recognition algorithm, as in Nefian *et al.* (2003).

Modeling others' intentions has been considered in social decision-making, see Falk *et al.* (2003) and Charness & Levine (2007). A method capable of capturing others' intentions, based on gathering information to infer their actions and model their possible goals and strategies, would help us in complementing the forecasting models of our social decision-making agent, and in supporting our agent in knowing whether to behave reciprocally or not, see Falk & Fischbacher

(2006).

Following Valtazanos & Ramamoorthy (2013) ideas, we should consider the possibility of including shape interaction concepts, through which we would provide our supported agent with the ability of influencing the state of other strategic agents within interactive situations. Our agent would not be allowed to force the other agent to perform the desire action, but shape the interaction indirectly by identifying those actions which can persuade the adversary to behave in accordance with the desire goal.

Through the described model in Chapter 3, we allow an agent to move from competitive to cooperative attitudes, and vice versa. Such model is based on a cooperativeness parameter which influences its attitude. Our affective model includes some affective factors, e.g. the mood, which may influence the agents behavior towards optimistic or pessimistic choices, see Bower (1991) and Wright & Bower (1992). Combining both ideas, we would allow an agent to modify its degree of cooperativeness based on the evolution of its mood. We would expect that, within a society, agents in a better mood would be willing to cooperate, whereas those in worse mood would behave more competitively.

We have contextualized our interactive model within two competitive scenarios in Section 3.3, and a cooperative one in Section 3.4. More interactive situations, involving different human-agent formations, levels of sophistication or interacting attitudes, may be studied to assess these models. For example, we may find situations in which some agents cooperate among them to support another robotic agent in its decision-making process when it is facing a user.

Within the affective model, we have made the proposed parametric choices based on trial and error. So that some work would be required to fix them in an optimal way to improve the performance of the model.

In order to make this model more complete, but still avoiding an increase on the complexity of the system, we shall consider as future work the addition of personality traits to influence the emotions and mood generation, and the behavioral model of our agent, see Ortony (2002). Other affective phenomena, as motivational states and attitudes, would also be considered. Adding more basic emotions, such as anger or boredom, would be interesting to keep interactions attractive. Anger may rise, e.g., when the distance between the expected outcome and the perceived outcome is high; and boredom may rise, e.g., when there is no variation on the utilities obtained or the expected utilities computed, for a certain number of loop iterations.

Within this work we have only considered those affective states simulated by the supported agent, but, to be realistic, we should not assume that the remaining individuals within the society are affectiveless agents. So that, through capturing others' facial expressions, body posture and gestures, or even the acoustic of their voice, we should infer others' affective states, see Picard (1997), improving social interactions and providing a more realistic experience for the user. Combining this with models of others' intentions could be taken into account to build highly sophisticated forecasting models.

Loewenstein & Lerner (2003) use the term *hot/cold empathy gap* referring to those situations in which we try to forecast how we will behave if we were in the opposite mood, where we tend to fail most of the times. Then, we could complement our forecasting models including such concept. To do so, we may use prospect theory ideas, see Kahneman & Tversky (1979), to explore how affective factors impact on the agent's forecasting system.

Emotion expression could be improved by combining emotions through mixing different colors in its RGB led. Moreover, we should explore the possibility of expressing emotions through the acoustic properties of its voice, see Bachorowski (1999).

Through this thesis we have implemented two models within the AiSoy1 robot: an ARA based basic model and its extension including affective factors. We may implement the interactive models to allow the AiSoy1 robot to cooperate, compete and autonomously interact with children, in order to study how they would behave under different realistic social situations. Extending the interactive models with affective factors would allow us to perform experiments in social environments to study how emotions may influence the social behavior of our autonomous agent within a society of agents and humans.

The AiSoy1 robot, with these models implemented in it, may be a very useful tool to be used within experimental game theoretic labs, testing how different decision models perform in their interaction with it. Apart from that, we consider that these models may be applied in several real environments like those described next. In edutainment situations, an AiSoy1 robot may be used to complement daily lessons at school or being used as a robotic tutor at home, to help the children do their homework. Recently, social robots have been proved to be particularly effective for mediating between therapists and children with autism spectrum disorders (ASD), due to their ability to express emotional states, as individuals with autism show diculties with cognitive and motor empathy but less clear dificulties with respect to emotional empathy. There is a growing need for new technologies that can assist the elderly in their daily living as people prefer to live in their own homes instead of being institutionalized in sheltered homes or nursery homes, when problems related to ageing appear.

Finally, models described through this thesis may be extended to be implemented within more complex robotic or non-robotic platforms. They would be surrounded by more complex environments, in which additional sensors may be needed to infer more complete and realistic sets of actions and environmental states. Provided with more sophisticated processors we may extend the limited memory of our agent to be able to forecast further steps forward, see Section 2.3.3. Motion-related actions would be considered if the incumbent actuators are available, extending considerably the set of available actions, possibly changing the environment. The preference model should also be modified depending on the platform in which models are going to be placed.

References

ADLER, M.R., DAVIS, A.B., WEHMAYER, R. & WORREST, R.W. (1989). Conflict resolution strategies for nonhierarchical distributed agents. In *Distributed Artificial Intelligence*, vol. 2, 139–161, Morgan Kaufmann Publishers Inc.

AISOYROBOTICS (2010). Aisoy1 robot. http://www.aisoy.es/.

ALIPRANTIS, C. & CHAKRABARTI, S. (2010). *Games and Decision Making*. Oxford University Press.

ANDREONI, J. & MILLER, J. (2002). Giving according to garp: An experimental test of the consistency of preferences for altruism. *Econometrica*, **70**, 737–753.

ASADA, M., STONE, P., KITANO, H., DROGOUL, A., DUHAUT, D., VELOSO, M., ASAMAS, H. & SUZUKI, S. (1998). The robocup physical agent challenge: Goals and protocols for phase i. In *RoboCup-97: Robot Soccer World Cup I*, 42–61, Springer.

AUSTIN, J. (1962). *How to Do Things with Words*. Oxford: Clarendon Press.

BACHOROWSKI, J.A. (1999). Vocal expression and perception of emotion. *Current directions in psychological science*, **8**, 53–57.

BALDI, P. & ITTI, L. (2010). Of bits and wows: a bayesian theory of surprise with applications to attention. *Neural Networks*, **23**, 649–666.

BANKS, D., PETRALIA, F. & WANG, S. (2011). Adversarial risk analysis: Borel games. *Applied Stochastic Models in Business and Industry*, **27**, 72–86.

Bechara, A., Damasio, H. & Damasio, A.R. (2000). Emotion, decision making and the orbitofrontal cortex. *Cerebral Cortex*, **10**, 295–307.

Beedie, C., Terry, P.C. & Lane, A. (2005). Distinctions between emotion and mood. *Cognition and Emotion*, **19**, 847–878.

Beigi, H. (2011). *Fundamentals of Speaker Recognition*. Springer.

Bellifemine, F., Claire, G. & Greenwood, D. (2007). *Developing multi-agent systems with JADE*. John Wiley and Sons.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

Beni, G. & Wang, J. (1989). Swarm intelligence in cellular robotic systems. In *Proceed. NATO Advanced Workshop on Robots and Biological Systems*.

Bolles, R.C. & Fanselow, M.S. (1980). A perceptual-defensive-recuperative model of fear and pain. *Behavioral and Brain Sciences*, **3**, 291–301.

Bolton, G.E. (1991). A comparative model of bargaining: Theory and evidence. *The American Economic Review*, 1096–1136.

Bolton, G.E. & Ockenfels, A. (2000). Erc: A theory of equity, reciprocity, and competition. *American economic review*, 166–193.

Bonabeau, E., Thraulaz, G., Deneubourg, J., Aron, S. & Camazine, S. (1997). Self-organization in social insects. *Trends in Ecological Evolution*, **12**, 188–193.

Bower, G.H. (1991). Mood congruity of social judgments. *Emotion and social judgments*, 31–53.

Breazeal, C. (1998). A motivational system for regulating human-robot interaction. In *Proceedings AAAI/IAAI*, 54–61.

Breazeal, C. (2003). Emotion and sociable humanoid robots. *International Journal of Human-Computer Studies*, **59**, 119–155.

Breazeal, C. & Scassellati, B. (1999). How to build robots that make friends and influence people. In *Proc. IROS99, Kyonjiu, Korea*, 858–863.

Breazeal, C. & Scassellati, B. (2000). Infant-like social interactions between a robot and a human caregiver. *Adaptive Behavior*, **8**, 49–74.

Breazeal, C., Brooks, A., Gray, J., Hoffman, G., Kidd, C., Lee, H., Lieberman, J., Lockerd, A. & Chilongo, D. (2004a). Tutelage and collaboration for humanoid robots. *International Journal of Humanoid Robotics*, **1**, 315–348.

Breazeal, C., Hoffman, G. & Lockerd, A. (2004b). Teaching and working with robots as a collaboration. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 1030–1037.

Brehm, J.W. & Miron, A.M. (2006). Can the simultaneous experience of opposing emotions really occur? *Motivation and Emotion*, **30**, 13–30.

Broekens, J., Heerink, M. & Rosendal, H. (2009). Assistive social robots in elderly care: a review. *Gerontechnology*, **8**, 94–103.

Brooks, R.A. & Flynn, A.M. (1989). *Fast, cheap and out of control*. Tech. rep., DTIC Document.

Brown, G.W. (1951). Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, **13**, 374–376.

Brown, R.G. (1963). *Smoothing Forecasting and Prediction of Discrete Time Series*. Englewood Cliffs, NJ: Prentice-Hall.

Bruce, A., Nourbakhsh, I. & Simmons, R. (2002). The role of expressiveness and attention in human-robot interaction. In *Proceedings International Conference on Robotics and Automation*, vol. 4, 4138–4142.

Bui, T. (2004). *Creating Emotions and Facial Expressions for Embodied Agents*. Ph.D. thesis, University of Twente.

Busemeyer, J.R., Dimperio, E. & Jessup, R.K. (2006). *Integrating emotional processes into decision-making models*. Oxford University Press, New York, NY.

Camerer, C.F. (2003). *Behavioral Game Theory: Experiments on Strategic Interaction*. Princeton, NJ: Princeton University Press.

Camerer, C.F. (2008). *Neuroeconomics: Decision Making and the Brain*, chap. Behavioral Game Theory and the Neural Basis of Strategic Choice. Academic Press.

Camerer, C.F. & Fehr, E. (2006). When does "economic man" dominate social behavior? *Science*, **311**, 47–52.

Cañamero, D. (1997). Modeling motivations and emotions as a basis for intelligent behavior. In *Proceedings of the first international conference on Autonomous agents*, 148–155.

Cañamero, L. & Fredslund, J. (2001). I show you how i like you-can you read it in my face? *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **31**, 454–459.

Carnegie Melon, U. (2013). Cmu sphinx open source toolkit for speech recognition. http://cmusphinx.sourceforge.net/.

Charness, G. & Levine, D.I. (2007). Intention and stochastic outcomes: An experimental study. *The Economic Journal*, **117**, 1051–1072.

Clemen, R.T. & Reilly, T. (2004). *Making Hard Decisions with Decision Tools*. Duxbury: Pacific Grove, CA.

Clouse, J. (1996). *On integrating apprentice learning and reinforcement learning*. Ph.D. thesis, University of Massachusetts, Department of Computer Science.

Clyde, M. & George, E.I. (2004). Model uncertainty. *Statistical Science*, **19**, 81–94.

Cohen, J.D. (2005). The vulcanization of the human brain: A neural perspective on interactions between cognition and emotion. *Journal of Economic Perspectives*, **19**, 3–24.

Cohen, P.R. & Levesque, H.J. (1988). *Rational interaction as the basis for communication*. Tech. rep., DTIC Document.

Cohen, P.R. & Perrault, C.R. (1979). Elements of a plan-based theory of speech acts. *Cognitive science*, **3**, 177–212.

Damasio, A., Grabowski, T.J., Bechara, A., Damasio, H., Ponto, L., Parvizi, J. & Hichwa, R. (2000). Subcortical and cortical brain activity during the feeling of self-generated emotions. *Nature Neuroscience*, **3**, 1049–1056.

Damasio, A.R. (1994). *Descartes Error: Emotion, Reason, and the Human Brain*. New York: G.P. Putnam.

Darwin, C. (1782). *The expression of the emotions in man and animals*. Oxford University Press.

Davidson, R.J. (1994). *Nature of Emotion: Fundamental Questions*, chap. On Emotion, Mood and Related Affective Constructs. Ekman P., Editor, Oxford Unviversity Press: New York.

Díaz, J.G., Jurado, I.G. & Fiestras-Janeiro, M.G. (2010). *An introductory course on mathematical game theory*, vol. 115. AMS Bookstore.

Dufwenberg, M. & Kirchsteiger, G. (2004). A theory of sequential reciprocity. *Games and economic behavior*, **47**, 268–298.

Ekman, P. (1992). Are there basic emotions? *Psychological Review*, **99**, 550 – 553.

Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, **4**, 5–60.

Ekman, P. (2004). *Emotions Revealed: Recognizing Faces and Feelings to Improve Communication and Emotional Life*. Henry Holt and Co.

Ekman, P. & Friesen, W. (1977). *Facial Action Coding System*. Consulting Psychologists Press.

Ekman, P. & Friesen, W.V. (1975). *Unmasking the Face: A Guide to Recognizing Emotions from Facial Cues*. Englewood Cliffs, NJ: Prentice Hall.

El-Nasr, M.S., Yen, J. & Ioerger, T.R. (2000). Flame: Fuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-Agent Systems*, **3**, 219–257.

Elliott, C.D. (1992). *The affective reasoner: a process model of emotions in a multi-agent system*. Ph.D. thesis, Northwestern University.

Endriss, U. (2006). Monotonic concession protocols for multilateral negotiation. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 392–399.

Ephrati, E. & Rosenschein, J.S. (1993). Multi-agent planning as a dynamic search for social consensus. In *Proceedings of IJCAI*, vol. 93, 423–429.

Falk, A. & Fischbacher, U. (2006). A theory of reciprocity. *Games and Economic Behavior*, **54**, 293–315.

Falk, A., Fehr, E. & Fischbacher, U. (2003). On the nature of fair behavior. *Economic Inquiry*, **41**, 20–26.

Fehr, E. (2008). *Neuroeconomics: Decision Making and the Brain*, chap. Social Preferences and the Brain. Academic Press.

Fehr, E. & Schmidt, K.M. (1999). A theory of fairness, competition and cooperation. *The Quarterly Journal of Economics*, **114**, 817–868.

Fehr, E. & Schmidt, K.M. (2006). The economics of fairness, reciprocity and altruism–experimental evidence and new theories. *Handbook of the Economics of Giving, Altruism and Reciprocity*, **1**, 615–691.

Fehr-Duda, H., Epper, T., Bruhin, A. & Schubert, R. (2011). Risk and rationality: The effects of mood and decision rules on probability weighting. *Journal of Economic Behavior & Organization*, **78**, 14 – 24.

Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McKay, D., Shapiro, S., McGuire, J., Pelavin, R. & Beck, C. (1994). *Specification of the KQML Agent-Communication Language*. Tech. rep.

Fiori, M., Lintas, A., Mesrobian, S. & Villa, A.E. (2013). Effect of emotion and personality on deviation from purely rational decision-making. In *Decision Making and Imperfection*, 129–161, Springer.

FIPA (1999). *Specification part 2, Agent Communication Language*. Tech. rep.

Fischer, F., Rovatsos, M. & Weiss, G. (2006). Adaptiveness in agent communication: Application and adaptation of conversation patterns. In *Agent Communication II*, 211–226, Springer.

Fong, T., Nourbakhsh, I. & Dautenhahn, K. (2003). A survey of socially interactive robots. *Robotics and autonomous systems*, **42**, 143–166.

Freimer, M. & Yu, P.L. (1976). Some new results on compromise solutions for group decision problems. *Management Science*, **22**, 688–693.

Frijda, N.H. (1986). *The Emotions*. Cambridge University Press.

Frijda, N.H. (1994). Varieties of affect: Emotions and episodes, moods, and sentiments. *The Nature of Emotions: Fundamental Questions*, 197–202.

Fujita, M. & Kitano, H. (1998). Development of an autonomous quadruped robot for robot entertainment. *Auton. Robots*, **5**, 7–18.

Fujita, Y. (2002). Personal robot papero. *J. Robot. Mechatron.*, **14**, 60–63.

Gal, Y., Grosz, B., Kraus, S., Pfeffer, A. & Shieber, S. (2010). Agent decision-making in open mixed networks. *Artificial Intelligence*, **174**, 1460–1480.

Gal, Y., Kraus, S., Gelfand, M., Khashan, H. & Salmon, E. (2011). An adaptive agent for negotiating with people in different cultures. *ACM Transactions on Intelligent Systems and Technology (TIST)*, **3**, 8.

GALLIERS, J.R. (1988). *A theoretical framework for computer models of cooperative dialogue, acknowledging multi-agent conflict.*. Ph.D. thesis, Open University.

GALLIERS, J.R. (1990). The positive role of conflict in cooperative multi-agent systems. *Demazeau & Müller 1990*, 33–46.

GIBBONS, R. (1992). *A primer in game theory*. Harvester Wheatsheaf Hertfordshire.

GLIMCHER, P.W., CAMERER, C., POLDRACK, R.A. & FEHR, E. (2008). *Neuroeconomics: Decision Making and the Brain*. Academic Press.

GOLDBERG, L.R. (1990). An alternative "description of personality": the big-five factor structure. *Journal of personality and social psychology*, **59**, 1216.

GRATCH, J. (2000). Émile: Marshalling passions in training and education. In *Proceedings of the Fourth International Conference on Autonomous Agents*, 325–332.

GRATCH, J. & MARSELLA, S. (2004). A domain-independent framework for modeling emotion. *Journal of Cognitive Systems Research*, **5**, 269–306.

GROSZ, B.J., KRAUS, S., TALMAN, S., STOSSEL, B. & HAVLIN, M. (2004). The influence of social dependencies on decision-making: Initial investigations with a new game. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, vol. 2, 782–789.

GUY, T.V., KARNY, M. & WOLPERT, D. (2013). *Decision Making and Imperfection*. Springer.

HARLE, K. & SANFEY, A. (2007). Sadness biases social economic decisions in ultimatum game. *Emotion*, **7**, 876–881.

HARSANYI, J.C. (1967). Games with incomplete information played by "bayesian" players. *Management Science*, **14**, 159–182.

Harsanyi, J.C. (1982). Subjective probability and the theory of games: Comments on kadane and larkey's paper. *Management Science*, **28**, 113–120.

Hewitt, R. (2007). Seeing with opencv, part 4: Face recognition with eigenface. http://www.cognotics.com/opencv/servo_2007_series/part_4/index.html.

Hoeting, J., Madigan, D., Raftery, A. & Volinsky, C. (1999). Bayesian model averaging: A tutorial. *Statistical Science*, **4**, 382–417.

Huget, M.P. (2003). *Communication in Multiagent Systems*. Springer-Verlag.

Itti, L. & Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research*, **49**, 1295 – 1306.

Izard, C.E. & Ackerman, B.P. (2000). Motivational, organizational, and regulatory functions of discrete emotions. *Handbook of emotions*, **2**, 253–264.

Jim, K.C. & Giles, C.L. (2001). How communication can improve the performance of multi-agent systems. In *Proceedings of the fifth international conference on Autonomous agents*, 584–591.

Kadane, J.B. & Larkey, P.D. (1982). Subjective probability and the theory of games. *Management Science*, **28**, 113–120.

Kahneman, D. & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica: Journal of the Econometric Society*, 263–291.

Kalai, E. & Smorodinsky, M. (1975). Other solutions to Nash's bargaining problem. *Econometrica*, **43**, 513–518.

Keltner, D., Ekman, P., Gonzaga, G.C. & Beer, J. (2003). *Facial Expression of Emotion*. Oxford University Press.

Kirby, R., Forlizzi, J. & Simmons, R. (2010). Affective social robots. *Robotics and Autonomous Systems*, **58**, 322–332.

Kirchsteiger, G. (1994). The role of envy in ultimatum games. *Journal of economic behavior & organization*, **25**, 373–389.

KONE, M., SHIMAZU, A. & NAKAJIMA, T. (2000). The state of the art in agent communication languages. *Knowledge and Information Systems*, **2**, 259–284.

KOZIMA, H. & YANO, H. (2001). A robot that learns to communicate with human caregivers. In *Proceedings of the First International Workshop on Epigenetic Robotics*, 47–52.

KOZIMA, H., MICHALOWSKI, M.P. & NAKAGAWA, C. (2009). Keepon, a playful robot for research, therapy, and entertainment. *International Journal of Social Robotics*, **1**, 3–18.

KULLBACK, S. & LEIBLER, R.A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, **22**, 49–86.

LANDELS, J.G. (1979). Water-clocks and time measurement in classical antiquity. *Endeavour*, **3**, 32–37.

LAZARUS, R. (1991a). *Emotion and Adaptation*. Oxford University Press.

LAZARUS, R.S. (1991b). Progress on a cognitive-motivational-relational theory of emotion. *American Psychologist*, **46**, 819–834.

LEVINE, D.K. (1998). Modeling altruism and spitefulness in experiments. *Rev. Econ. Dynamics*, **1**, 593–622.

LIEMHETCHARAT, S. & VELOSO, M. (2012). Modeling and learning synergy for team formation with heterogeneous agents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, 365–374.

LIPPMAN, S. & MCCARDLE, K. (2012). Embedded nash bargaining: Risk aversion and impatience. *Decision Analysis*, **9**, 31–41.

LOEWENSTEIN, G. (2000). Emotions in economic theory and economic behavior. *American Economic Review*, **90**, 426–432.

LOEWENSTEIN, G. & LERNER, J.S. (2003). *The role of affect in decision making*, 619–642. In R. Davidson, K. Scherer and H. Goldsmith (eds.), Handbook of Affective Science, Oxford: Oxford University Press.

Mariotti, M. (1998). Nash bargaining theory when the number of alternatives can be finite. *Social Choice and Welfare*, **15**, 413–421.

Maslow, A.H. (1943). A theory of human motivation. *Psychological Review*, **50**, 370–396.

McCrae, R.R. & Costa, P.T. (1997). Personality trait structure as a human universal. *American Psychologist*, **52**, 509–516.

Mellers, A.B., Schwatz, A., Ho, K. & Ritov, I. (1997). Decision affect theory: Emotional reactions to the outcomes of risky options. *Psychological Science*, **8**, 423–429.

Menache, I. & Ozdaglar, A. (2011). Network games: Theory, models, and dynamics. *Synthesis Lectures on Communication Networks*, **4**, 1–159.

Michaud, F., Pirjanian, P., Audet, J. & Létourneau, D. (2000). Artificial emotion and social robotics. In *Distributed Autonomous Robotic Systems*, 121–130, Springer.

Moshkina, L.V. (2011). *An Integrative Framework of Time-Varying Affective Robotic Behavior*. Ph.D. thesis, Georgia Institute of Technology.

Myerson, R. (1991). *Game theory: analysis of conflict*. Harvard University Press.

Nagahisa, R. & Tanaka, M. (2002). An axiomatization of the kalai-smorodinsky solution when the feasible sets can be finite. *Social Choice and Welfare*, **19**, 751–761.

Nash, J. (1950). The bargaining problem. *Econometrica*, **18**, 155–162.

Nash, J. (1951). Non-cooperative games. *The Annals of Mathematics*, **54**, 286–295.

Nass, C.I., Moon, Y., Morkes, J., Kim, E.Y. & Fogg, B. (1997). Computers are social actors: a review of current research, human values and the design of computer technology. *Center for the Study of Language and Information, Stanford, CA*, 137–162.

NAZ, K. & EPPS, H. (2004). Relationship between color and emotion: a study of college students. *College Student Journal*, **38**, 396–405.

NEFIAN, A.V., LIANG, L.H., FU, T. & LIU, X.X. (2003). A bayesian approach to audio-visual speaker identification. In *Audio-and Video-Based Biometric Person Authentication*, 761–769, Springer.

NICOLESCU, M. & MATARIC, M.J. (2003). Natural methods for robot task learning: instructive demonstrations, generalization and practice. *Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 241–248.

NISAN, N., ROUGHGARDEN, T., TARDOS, E. & VAZIRANI, V.V. (2007). *Algorithmic Game Theory*. Cambridge University Press.

NOWAK, M.A., PAGE, K.M. & SIGMUND, K. (2000). Fairness versus reason in the ultimatum game. *Science*, **289**, 1773–1775.

O'DOHERTY, J.P. (2004). Reward representations and reward-related learning in the human brain: insights from neuroimaging. *Current Opinion in Neurobiology*, **14**, 769 – 776.

ORTONY, A. (2002). *Emotions in humans and artifacts*, chap. On making believable emotional agents believable. In Trappl, et al. (Editors).

ORTONY, A., CLORE, G.L. & COLLINS, A. (1988). *The Cognitive Structure of Emotions*. Cambridge University Press.

PANKSEPP, J. (2000). *Handbook of Emotions*, chap. Emotions As Natural Kinds within the Mammalian Brain. M. Lewis, Haviland-Jones, J.M., editors.

PATÈ-CORNELL, M.E. & GUIKEMA, S.O. (2002). Probabilistic modeling of terrorist threats: A systems analysis approach to setting priorities among countermeasures. *Military Operation Research*, **7**, 5–23.

PAVLOV, I.P. (1927). *Conditioned reflexes*. Dover Publications.

Phillips, M. & Veloso, M. (2009). Robust supporting role in coordinated two-robot soccer attack. In *RoboCup 2008: Robot Soccer World Cup XII*, 235–246, Springer.

Picard, R.W. (1997). *Affective Computing*. Cambridge, MA: MIT Press.

Pineau, J., Montemerlo, M., Pollack, M., Roy, N. & Thrun, S. (2003). Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems*, **42**, 271–281.

Plutchik, R. (2001). The nature of emotions. *American Scientist*, **89**, 344–350.

Price, B. & Boutilier, C. (2003). Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research (JAIR)*, **19**, 569–629.

Rabin, M. (1993). Incorporating fairness into game theory and economics. *American Economic Review*, **83**, 1281–1302.

Raiffa, H. (1982). *The Art and Science of Negotiation*. Cambridge, Massachusetts: Press of Harvard University Press.

Raiffa, H., Richardson, J. & Metcalfe, D. (2007). *Negotiation Analysis: The Science and Art of Collaborative Decision Making*. Cambridge, Massachusetts: Press of Harvard University Press.

Raspberry Pi, F. (2011). Raspberri pi model b. http://www.raspberrypi.org/.

Rázuri, J.G., Esteban, P.G. & Ríos Insua, D. (2012). An adversarial risk analysis model for an autonomous imperfect decision agent. In *Decision Making and Imperfection*, 163–187, T. V. Guy et al.(Eds).

Riley, P. (2005). *Coaching: Learning and Using Environment and Agent Models for Advice*. Ph.D. thesis, computer Science Dept., Carnegie Mellon University.

Ríos Insua, D. & Salewicz, K. (1995). The operation of kariba lake: a multiobjective decision analysis. *Journal of Multicriteria Decision Analysis*, **4**, 203–222.

Ríos Insua, D., Ríos, J. & Banks, D. (2009). Adversarial risk analysis. *Journal of the American Statistical Association*, **104**, 841–854.

Ríos Insua, D., Ruggeri, F. & Wiper, M. (2012). *Bayesian Analysis of Stochastic Process Models*. Wiley.

Rogers, B.W., Palfrey, T.R. & Camerer, C.F. (2009). Heterogeneous quantal response equilibrium and cognitive hierarchies. *Journal of Economic Theory*, **144**, 1440–1467.

Roseman, I. (1984). Cognitive determinants of emotion: A structural theory. *Review of Personality and Social Psychology: Emotions, relationships, and health*, **5**, 11–36.

Rosenschein, J.S. (1994). *Rules of encounter: designing conventions for automated negotiation among computers*. MIT Press.

Roth, M., Vail, D. & Veloso, M. (2003). A real-time world model for multi-robot teams with high-latency communication. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2494 – 2499.

Roth, M., Simmons, R. & Veloso, M. (2005). Reasoning about joint beliefs for execution-time communication decisions. In *Proceedings of International Joint Conference on Autonomous Agents and Multi Agent Systems*.

Rothkopf, M.H. (2007). Decision analysis: The right tool for auctions. *Decision Analysis*, **4**, 167–172.

Russell, J.A. (2003). Core affect and the psychological construction of emotion. *Psychological Review*, **110**, 145–172.

Russell, J.A. & Barrett, L.F. (1999). Core affect, prototypical emotional episodes, and other things called emotion: dissecting the elephant. *Journal of personality and social psychology*, **76**, 805–819.

Sanfey, A. & Dorris, M. (2008). *Neuroeconomics: Decision Making and the Brain*, chap. Games in Humans and Non-human Primates: Scanners to Single Units. Academic Press.

Schelling, T.C. (1960). *The Strategy of Conflict*. Cambridge: Harvard University Press.

Searle, J.R. (1969). *Speech acts: An essay in the philosophy of language*, vol. 626. Cambridge university press.

Sloman, A. (2002). *How many separately evolved emotional beasties live within us*, chap. Emotions in humans and artifacts. In Trappl, R. and Petta, P. and Payr, S., editors, MIT Press, Cambridge, MA.

Sribhashyam, S. & Montibeller, G. (2012). Modeling state - dependent priorities of malicious agents. *Decision Analysis*, **9**, 172–185.

Stahl, D.O. & Wilson, P.W. (1995). On players models of other players: Theory and experimental evidence. *Games and Economic Behavior*, **10**, 218–254.

Stebbins, G. (1886). *Delsarte system of dramatic expression*. ES Werner (New York).

Thomson, W. (1981). Nash's bargaining solution and utilitarian choice rules. *Econometrica: Journal of the Econometric Society*, 535–538.

Thomson, W. (1994). *Cooperative models of bargaining*, 1237 – 1284. In Elsevier (eds.), Handbook of Game Theory with Economic Applications.

Thomson, W. (2009). Bargaining and the theory of cooperative games: John nash and beyond. RCER Working Papers 554, University of Rochester - Center for Economic Research (RCER).

Turing, A.M. (1937). On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, **42**, 230–265.

TURING, A.M. (1950). Computing machinery and intelligence. *Mind LIX*, **236**, 433–460.

VALTAZANOS, A. & RAMAMOORTHY, S. (2013). Bayesian interaction shaping: learning to influence strategic interactions in mixed robotic domains. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, AAMAS '13, 63–70, International Foundation for Autonomous Agents and Multiagent Systems.

VAN BREEMEN, A.J. (2004). Animation engine for believable interactive user-interface robots. In *Proceedings of Intelligent Robots and Systems (IROS)*, vol. 3, 2873–2878, IEEE.

VELÁSQUEZ, J. (1998). A computational framework for emotion-based control. In *Proceedings of the Workshop on Grounding Emotions in Adaptive Systems; International Conference on SAB*, 62–67.

VELÁSQUEZ, J.D. (1997). Modeling emotions and other motivations in synthetic agents. In *Proceedings of AAAI97*.

VELÁSQUEZ, J.D. (1998). When robots weep: Emotional memories and decision-making. In *Proceedings of AAAI98*, 8.

VON NEUMANN, J. & MORGENSTERN, O. (1944). *Theory of Games and Economic Behavior*. Princeton, NJ, Princeton University Press.

WADA, K. & SHIBATA, T. (2007). Living with seal robots-its sociopsychological and physiological influences on the elderly at a care house. *Robotics, IEEE Transactions on*, **23**, 972–980.

WANG, S. & BANKS, D. (2011). Network routing for insurgency: An adversarial risk analysis framework. *Naval Research Logistics (NRL)*, **58**, 595–607.

WEERD, H., VERBRUGGE, R. & VERHEIJ, B. (2014). Agent-based models for higher-order theory of mind. In B. Kamiski & G. Koloch, eds., *Advances in Social Simulation*, vol. 229 of *Advances in Intelligent Systems and Computing*, 213–224, Springer Berlin Heidelberg.

WEST, M. & HARRISON, P.J. (1997). *Bayesian Forecasting and Dynamic Models*. New York: Springer.

WILCOX, B. (2013). Chat script. http://chatscript.sourceforge.net/.

WILLOWGARAGE (2008). Robot operative system. http://www.ros.org/.

WOOLDRIDGE, M. (2008). *An introduction to multiagent systems*. Wiley. com.

WOOLDRIDGE, M. & JENNINGS, N.R. (1997). Formalizing the cooperative problem solving process. *Synthese Library*, 143–162.

WRIGHT, W.F. & BOWER, G.H. (1992). Mood effects on subjective probability assessment. *Organizational behavior and human decision processes*, **52**, 276–291.

WU, H. (2007). *Finite Bargaining Problems*. Ph.D. thesis, Georgia State University.

YU, P.L. (1973). A class for solutions for group decision problems. *Management Science*, **19**, 936–946.

ZHANG, X. & LESSER, V. (2007). Meta-level coordination for solving negotiation chains in semi-cooperative multi-agent systems. In *Proceedings of AAMAS*.

ZHAO, W., CHELLAPPA, R., PHILLIPS, P.J. & ROSENFELD, A. (2003). Face recognition: A literature survey. *ACM Comput. Surv.*, **35**, 399–458.