



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Extensión de Jason para implementar agentes normativos emocionales

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Karen Yadira Lliguin León

Tutor: Estefanía Argente Villaplana
Vicente Juan Botti Navarro

Curso 2018-2019

Agradecimientos

"A mi familia y mi tutora"

Resum

La majoria de les decisions que prenen les persones incloses les econòmiques, es basen en gran mesura en consideracions normatiu-afectives, no només pel que respecta a la selecció d'objectius, sinó també dels mitjans. No obstant això, encara que les emocions són inherents al comportament humà i també són rellevants quan es tracta dels processos de presa de decisions, la relació entre normes i emocions gairebé no s'ha considerat en el camp (o bé àrea) dels sistemes multiagent, i la majoria dels sistemes multiagents normatius no les tenen en compte com una variable per al seu càlcul. Així, molts sistemes normatius modelen agents que realitzen el raonament pràctic sense tenir en compte les emocions de l'agent. Dins d'aquest marc, aquest treball de fi de grau proposa una extensió del llenguatge de programació de sistemes multiagent Jason, que permeti implementar un agent normatiu emocional (NEA) capaç de manejar tant normes com emocions. Per a això, s'analitzen els avantatges d'incloure emocions dins d'un sistema normatiu i com les emocions i les normes s'afecten entre si. En aquest treball es realitza una revisió de la treball fet en aquest camp fins ara i es presenta una proposta d'un model normatiu i emocional propis, que s'implementa com una extensió en Jason i finalment es presenta un cas d'estudi senzill per mostrar les aportacions dels agents NEA.

Paraules clau: Emocions, Normes, Jason, Sistemes normatius, BDI Agents, Raonament normatiu

Resumen

La mayoría de las decisiones que toman las personas, incluidas las económicas, se basan en gran medida en consideraciones normativo-afectivas, no solo en lo que respecta a la selección de objetivos, sino también de los medios. Sin embargo, aunque las emociones son inherentes al comportamiento humano y también son relevantes cuando se trata de los procesos de toma de decisiones, la relación entre normas y emociones apenas se ha considerado en el campo (o bien área) de los sistemas multiagente, y la mayoría de los sistemas multiagentes normativos no las toman en cuenta como una variable para su cálculo. Así, muchos sistemas normativos modelan agentes que realizan el razonamiento práctico sin tener en cuenta las emociones del agente. Dentro de este marco, este trabajo de fin de grado propone una extensión del lenguaje de programación de sistemas multiagente Jason, que permita implementar un agente normativo emocional (NEA) capaz de manejar tanto normas como emociones. Para ello, se analizan las ventajas de incluir emociones dentro de un sistema normativo y cómo las emociones y las normas se afectan entre sí. En este trabajo se realiza una revisión del trabajo realizado en este campo hasta ahora y se presenta una propuesta de un modelo normativo y emocional propios, que se implementa como una extensión en Jason y por último se presenta un caso de estudio sencillo para mostrar las aportaciones de los agentes NEA.

Palabras clave: Emociones, Normas, Jason, Sistemas normativos, Agentes BDI, Razonamiento normativo

Abstract

Most people's choices, including economic ones, are largely based on normative-affective considerations, not only with regard to the selection of goals but also of means. However, although emotions are inherent in human behavior, and they are also relevant when dealing with the decision-making processes, the relationship between norms and emotions has hardly been considered in the multiagent field, and most normative multi-agent systems do not take

emotions into account, as a variable for their computation. Thus, many normative systems model agents that perform practical reasoning without taking into account the emotions of the agent. Within this framework, this end-of-degree project proposes an extension for the multi-agent system programming language Jason, that will allow the implementation of an emotional normative agent (NEA) capable of dealing with both norms and emotions. To do this, we analyze the advantages of including emotions within a normative system and how emotions and norms affect each other. In this work, a review of the work done so far in this field is carried out, and we present a proposal for a normative model as well as an emotional model, which is implemented as an extension in Jason and finally a simple case study is presented to show the contributions of NEA agents.

Key words: Emotions, Norms, Jason, Normative systems, BDI Agents, Normative reasoning

Índice general

Índice general	V
Índice de figuras	VII
Índice de tablas	VII
<hr/>	
1 Introducción	1
1.1 Motivación	2
1.2 Objetivos	2
1.3 Relación del trabajo desarrollado con los estudios cursados	3
1.4 Estructura de la memoria	3
2 Estado del arte	5
2.1 Sistemas multiagente	5
2.1.1 Agente Inteligente	5
2.1.2 Arquitectura BDI	6
2.1.3 Jason	7
2.2 Sistemas multiagente normativos	11
2.2.1 Normas en Sistemas Multiagente	11
2.2.2 Sistemas multiagente normativos	13
2.3 Sistemas multiagente normativos emocionales	19
2.3.1 Emociones en Sistemas Multiagentes	20
2.3.2 Relación entre normas y emociones	21
2.3.3 Sistemas normativos emocionales	24
2.4 Discusión	28
3 NEA: Agente normativo emocional	31
3.1 Introducción	31
3.2 Arquitectura de un agente normativo emocional: NEA	33
3.2.1 Modelo normativo	33
3.2.2 Modelo emocional	35
3.2.3 Ciclo de un agente NEA	37
3.2.4 Actualización de la base de emociones (EBA)	38
3.2.5 Actualización del estado emocional del agente (EA)	38
3.2.6 Razonamiento normativo	41
3.3 Discusión	43
4 Caso de estudio	45
4.1 Caso de estudio	45
4.1.1 Introducción	45
4.1.2 Caso de estudio general: estación de taxis	46
4.1.3 Caso particular: estación con tres taxis	47
4.2 Implementación	48
4.2.1 Código del taxista	48

4.2.2	Modificaciones de funciones de Jason	50
4.3	Discusión	51
5	Conclusiones	53
5.1	Aportaciones y limitaciones	53
5.2	Dificultades y trabajo futuro	54
	Bibliografía	57

Índice de figuras

2.1	Ejemplo de sistema multiagente, donde los círculos representan a los agentes y las flechas sus interacciones	6
2.2	Agente inteligente	6
2.3	Ciclo de razonamiento de un agente en Jason	11
2.4	Modificaciones propuestas por Neto [8] en el ciclo de razonamiento Jason	16
2.5	Modificaciones propuestas por Lee [10] en el ciclo de razonamiento Jason	17
2.6	Modificaciones propuestas por Meneguzzi [11] en el ciclo de razonamiento Jason	18
2.7	Relación normas y emociones	22
2.8	Emociones utilizadas para el dilema de cumplimiento de las normas en Staller	25
2.9	Relación de las normas y emociones en la propuesta de Von Scheve et al.	26
2.10	Generación de emociones de acuerdo a la propuesta de Ahmad et al.	26
2.11	Generación de emociones de acuerdo a la propuesta de Ferreira et al.	27
3.1	Modelado de un sistema normativo emocional	35
3.2	Modelado de las emociones soportadas por un agente NEA	37
3.3	Ciclo de razonamiento propuesto para un agente normativo emocional (NEA) en Jason	37
3.4	Soporte de un agente NEA a las cuatro relaciones entre normas y emociones	43
4.1	Instantánea del entorno del caso de estudio	48
4.2	Creencias iniciales de un agente NEA taxista	49
4.3	Planes de un agente NEA taxista	50
4.4	Función de selección de evento de un agente NEA taxista	50
4.5	Función que realiza el razonamiento normativo del agente NEA taxista	51

Índice de tablas

2.1	Tipos de eventos activadores	9
2.2	Tipos de literales en el contexto de un plan	9
2.3	Clasificación de las normas	12
2.4	Sistemas multiagentes normativos	14
2.5	Comparación de la representación de la norma	14
2.6	Comparativa de los aspectos generales de cada propuesta	19
2.7	Estado del arte de Sistemas Normativos Emocionales	24

3.1	Emociones soportadas por un agente NEA	36
3.2	Registro de las e_{ENE} recibidas en el instante t	42
4.1	Tipos de Roles de taxistas	47

CAPÍTULO 1

Introducción

Hoy en día la resolución de muchos problemas complejos de diferentes ámbitos requiere de un trabajo colaborativo y procesos descentralizados. Los sistemas multiagentes (MAS¹) debido a sus características son buenos candidatos para el desarrollo de sistemas capaces de manejar este tipo de problemas. Un sistema multiagente se define como una colección de agentes, que son entidades computacionales que poseen autonomía para actuar e interactuar entre sí (cooperación, coordinación, negociación...) y con su entorno para conseguir un objetivo, que puede ser común o individual. Debido a la interacción existente entre los agentes es necesario un mecanismo de control para evitar conflictos y asegurar que el funcionamiento dentro del entorno sea correcto [2]. Este mecanismo de control se puede conseguir a través de las normas.

Los sistemas normativos han sido objeto de estudio de la informática estos últimos años [3] porque permiten controlar y regular las interacciones dentro de un entorno a través de las normas, que definen lo que está o no permitido.

Existe una gran cantidad de definiciones del concepto de "norma". Una de las razones es debido a que su investigación se encuentra dispersa en una amplia gama de disciplinas diferentes. Su objetivo principal es el de modelar la interacción entre los distintos miembros de la sociedad. Las normas pueden o no tener asociadas un premio o castigo en caso de cumplir o violar la norma para favorecer su cumplimiento. Por ello las normas que rigen un sistema normativo son una buena herramienta para controlar la autonomía de los agentes dentro de los sistemas multiagentes.

Se distinguen dos aproximaciones para implementar las normas en un sistema normativo. La primera consiste en implementarlas directamente en el agente, de modo que las normas se adoptan sistemáticamente. En consecuencia, el agente pierde cierta autonomía, que es una característica fundamental para él, al no disponer de mecanismos para razonar sobre las normas y decidir si cumplirlas o no. La segunda opción consiste en implementar los mecanismos adecuados con los que el agente pueda realizar un razonamiento normativo previo, para decidir si se adopta o rechaza la norma, respetando así la autonomía del agente.

La norma por tanto, sirve para regular el comportamiento de los agentes y, en su caso, influir en su toma de decisiones. No obstante, en el ámbito humano, la mayoría de las decisiones que toma la gente, incluidas las económicas, se basan en gran parte en consideraciones normativo afectivas, no sólo respecto a la elección tomada sino también al medio por el cual se obtiene. Sin embargo aunque las emociones son inherentes en el comportamiento de los seres humanos y son relevantes cuando se trata con procesos de toma de decisiones, poco se ha estudiado sobre un modelo de agente que incluya el factor emocional en el proceso de toma de

¹Multiagent system

decisión. Asimismo la relación entre las normas y las emociones apenas se ha considerado en el campo de agentes y la mayoría de sistemas multiagente normativos no la considera como una variable a tomar en cuenta.

Con respecto a las emociones, dentro de la inteligencia artificial la computación afectiva[4] desarrolla y estudia sistemas capaces de reconocer, procesar, interpretar y simular emociones humanas. La incorporación de las emociones en los agentes está relacionada con el razonamiento práctico, de manera que permite a los agentes interpretar y adaptar su comportamiento, dando una respuesta más apropiada.

1.1 Motivación

Las propuestas existentes analizadas sobre sistemas normativos emocionales en su mayoría imponen las normas en los agentes, es decir, las modelan de manera que los agentes no pueden razonar sobre ellas y no tienen más opción que cumplir las normas quedando limitada así la autonomía del agente. Por ello resulta interesante profundizar más en una propuesta de un sistema multiagente que englobe ambos: normas y emociones y además sea capaz de asegurar la libertad del agente para decidir sobre las normas. Así este trabajo propone el diseño e implementación de un sistema con agentes normativos emocionales (NEA²) que sea capaz de integrar tanto la parte normativa del sistema como la parte emocional asociada a cada agente.

Para ello se extiende el lenguaje de agentes *Jason* para incluir los conceptos de norma y de emoción, pudiendo así modelar un *agente normativo emocional* (NEA).

1.2 Objetivos

Este trabajo se centra en el análisis, diseño e implementación de un sistema normativo emocional cuyos agentes posean autonomía total, es decir que las normas no se encuentren inherentes en los agentes, sino que puedan razonar sobre ellas y para cuyo razonamiento se tengan también en mente las emociones de los agentes. Para abordar este objetivo general se irán afrontando los siguientes subobjetivos:

O.1 Análisis y revisión de los trabajos existentes sobre sistemas normativos.

O.1.1 Revisión del concepto de norma y clasificación de los diferentes tipos de normas.

O.1.2 Revisión de la plataforma Jason y su ciclo de razonamiento.

O.1.3 Identificar los retos existentes en el campo de los sistemas normativos.

O.2 Análisis y revisión de los trabajos más relevantes sobre los sistemas normativos emocionales.

O.2.1 Revisión del concepto de emoción y presentación de los sistemas emocionales principales.

O.2.2 Investigación sobre la incorporación de las emociones en agentes inteligentes.

O.2.3 Estudio de cómo se afectan mutuamente las emociones y normas además de cómo afectan al comportamiento del agente.

²*Normative Emotional Agent*

- O.2.4 Identificar los retos existentes en el campo de los sistemas normativos emocionales.
- O.3 Presentación e implementación de las características de la propuesta de NEA.
- O.4 Construir un caso de estudio específico en Jason y presentar los resultados obtenidos.

1.3 Relación del trabajo desarrollado con los estudios cursados

El contenido de este trabajo surge en primer lugar, del trabajo como becaria realizado dentro del grupo de investigación del GTI-IA, en concreto en el Departamento de Sistemas Informáticos y Computación. Es más, parte de este trabajo ha sido presentado en el Congreso AT 2017 y publicado en las actas de ese Congreso (libro Springer 2018). Gracias a ello, pude ondar más en el conocimiento de sistemas multiagentes normativos y emocionales, dentro de la rama de la inteligencia artificial. En 3º curso tuve mi primera aproximación hacia la Inteligencia Artificial, con la asignatura de Sistemas Inteligentes y el año siguiente, en 4º curso profundicé aún más gracias a la asignatura de Agentes Inteligentes. Respecto a las tecnologías empleadas en este trabajo, para implementar el entorno del caso de estudio se empleó el lenguaje de Java, que manejo con facilidad habiendo cursado las asignaturas de programación de este grado y mis propios conocimientos. Sin embargo para la implementación del agente NEA que se propone, se hace uso del lenguaje *JASON* por los beneficios que ofrece a la hora de crear agentes inteligentes. No obstante mi experiencia en el uso de este lenguaje era básico, debido a que mi primera aproximación a este lenguaje fue en cuarto curso la asignatura de Agentes Inteligentes. A pesar de eso durante el proceso de trabajo de fin de grado he podido aprender a manejarlo mucho mejor.

1.4 Estructura de la memoria

El presente trabajo está organizado como sigue:

- Capítulo 2: se presenta el estado del arte de los sistemas normativos y los sistemas emocionales. Además, se explican conceptos básicos sobre cada sistema y se presenta la plataforma Jason y su entorno de desarrollo.
- Capítulo 3: se detallan las características del agente normativo emocional (NEA) propuesto.
- Capítulo 4: se presenta el caso de estudio y los resultados obtenidos.
- Capítulo 5: se presentan las conclusiones generales y los retos existentes en este campo.

CAPÍTULO 2

Estado del arte

En este capítulo se presenta el concepto de sistema multiagente y sus componentes generales, así como conceptos básicos que se usan a lo largo del trabajo. También se presenta la arquitectura BDI¹ y la plataforma Jason. Además se expone una revisión del trabajo más relevante sobre sistemas normativos (en específico aquellos basados en Jason) y sobre sistemas normativos emocionales. Finalmente se muestran las conclusiones obtenidas.

2.1 Sistemas multiagente

Los sistemas multiagentes (MAS²) han sido objeto de estudio debido a sus características ya que son buenos candidatos para modelar sistemas distribuidos abiertos y heterogéneos. Es por esto por lo que el uso de este tipo de tecnología se ha incrementado, no sólo en ámbito académico sino también en el desarrollo e implementación de aplicaciones industriales [3]. Un MAS se define como un sistema compuesto por varios agentes que interactúan con el entorno y entre sí no sólo intercambiando información sino también realizando acciones sociales tales como la cooperación, la coordinación o la negociación [6] (ver figura 2.1). En un MAS los agentes o bien cooperan para conseguir una meta común o compiten entre ellos para conseguir sus metas personales.

2.1.1. Agente Inteligente

No existe consenso respecto a la definición de "agente" debido a su uso en un amplio rango de campos donde en cada uno se prioriza una u otra característica. Sin embargo se puede ofrecer una definición a grandes rasgos de lo que se entiende por *agente inteligente* dentro del campo de la informática. Un agente inteligente es visto como una entidad computacional situada en un entorno, generalmente con más agentes con los que es capaz de comunicarse, donde en base al conocimiento que posee actúa en consecuencia para alcanzar su objetivo marcado [6].

El agente interactúa con su entorno mediante sensores que recogen las percepciones sobre su entorno así como del resto de agentes del sistema. Después, mediante un mecanismo de inferencia, procesa las percepciones obtenidas y construye una respuesta usando la información que posee almacenada en la base de conocimiento, como se muestra en la Figura 2.2.

¹Belief-Desire-Intention

²Multiagent system

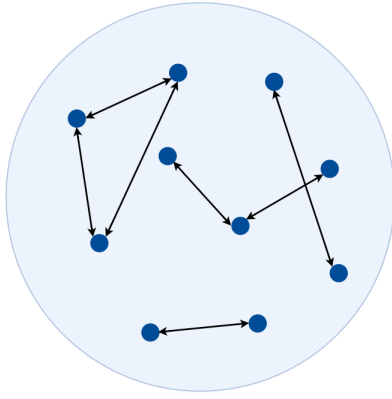


Figura 2.1: Ejemplo de sistema multiagente, donde los círculos representan a los agentes y las flechas sus interacciones

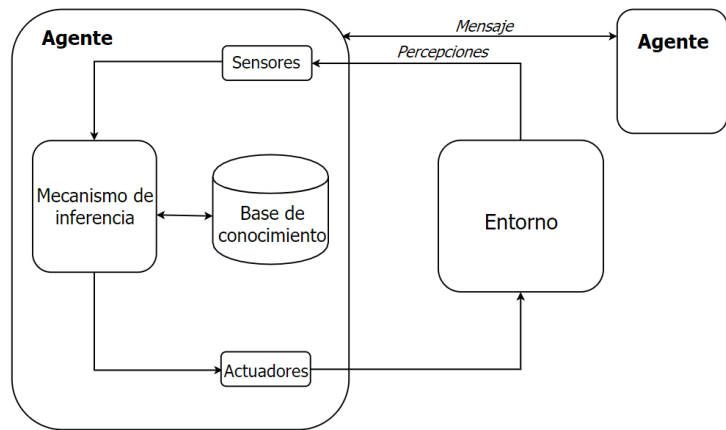


Figura 2.2: Agente inteligente

Wooldridge [6] destaca una serie de características que se consideran básicas en un agente inteligente. Dichas cualidades denotan ciertas propiedades que el agente debe cumplir:

- **Autonomía:** el agente debe tener la libertad para poder decidir las acciones a realizar para alcanzar las metas establecidas. De manera que tiene control sobre sus actos, sin ningún tipo de intromisión humana directa.
- **Reactividad:** el agente debe ser capaz de percibir su entorno e interactuar con él mediante sensores y actuadores, con el fin de que pueda responder a los cambios que ocurren adaptando sus acciones a los cambios que observa.
- **Proactividad:** el agente debe presentar un comportamiento emprendedor, generando y tratando de lograr metas, no ser conducido únicamente por los acontecimientos sino tomar iniciativa para actuar y controlar sus propios objetivos.
- **Sociabilidad:** el agente debe ser capaz de comunicarse e interactuar con el resto de agentes para así poder intercambiar información necesaria para poder tomar las decisiones, así como para cooperar y coordinar sus actividades con otros agentes, a fin de conseguir sus objetivos.

2.1.2. Arquitectura BDI

Existen varias propuestas de arquitecturas para modelar agentes [27]: agentes basados en lógica, agentes reactivos, arquitecturas por capas y agentes BDI. De entre ellos este trabajo se centra en el modelo BDI, que es uno de los más usados e importantes, además de ser el más aceptado para representar agentes capaces de realizar un razonamiento complejo. Está basado en la teoría de *razonamiento práctico* desarrollada por el filósofo Michael Bratman [53]. A diferencia de otros modelos previos que sólo consideraban las creencias y deseos del agente, este modelo toma en cuenta también las intenciones inmediatas y su función dentro del razonamiento práctico. Modela un agente racional donde las creencias, deseos e intenciones son la base del estado mental del agente. A continuación se detalla cada estado mental:

- Creencias: es toda aquella información interna que el agente tiene disponible acerca de él mismo, del entorno o de otros agentes. Puede obtenerlas a través de percepciones o de mensajes de otros agentes. Se corresponde con "lo que el agente piensa".
- Deseos: representan los posibles objetivos a los que el agente querría llegar o alcanzar, aunque esto no implica necesariamente que acabe actuando de acuerdo a ellos. Se podrían ver como las opciones que el agente está barajando. Se corresponde con "lo que el agente quiere".
- Intenciones: son los estados que el agente ha decidido poner como objetivos y trabajar para conseguirlos. Representan la opción elegida de entre las posibles. Se corresponde con "lo que el agente está haciendo".

El razonamiento práctico representa el proceso de razonamiento que sigue el agente, de acuerdo a las creencias, deseos e intenciones que tiene, para decidir qué acciones llevar a cabo. Consta a su vez de dos procesos: deliberación, donde se establecen las intenciones del agente; y planificación o *means-end reasoning*³, que decide el cómo conseguir llegar al fin establecido usando los medios disponibles (i.e., acciones con las que cuenta el agente).

El razonamiento práctico se lleva a cabo en cuatro fases: (i) en primer lugar el agente recoge nuevas percepciones del estado actual de su entorno y del resto de agentes, con las que actualiza sus creencias (se borran/añaden creencias) y seguidamente se generan los deseos, es decir, las opciones de las que dispone a partir de las creencias actualizadas e intenciones que tenía el agente; (ii) delibera para decidir qué intención alcanzar; (iii) mediante la planificación, se determina el plan adecuado para alcanzar esa intención; y (iv) se ejecuta el plan, realizando las acciones que lo integran.

2.1.3. Jason

Jason es una extensión del lenguaje de programación *AgentSpeak*[1] que permite implementar sistemas multiagentes. En este trabajo se ha decidido usar Jason ya que permite la implementación de agentes que siguen el modelo BDI, además de permitir la modificación por parte del usuario de muchas de sus herramientas, como será necesario en este caso para poder implementar la propuesta de este trabajo. Asimismo ofrece un buen mecanismo de comunicación entre agentes, que es necesario para la correcta interacción entre ellos. Los componentes básicos de un agente BDI representado en Jason son las creencias, los objetivos y los planes.

Creencias

Las creencias del agente se representan mediante literales⁴ que representan una propiedad de un objeto o individuo que el agente cree verdadero en el entorno en el que se encuentra. Dicho predicado tiene como opción especificar atributos y anotaciones, como por ejemplo: *tocar(david, cello)[años_tocando(10)]*. Esta creencia presenta dos atributos que indican que el individuo *david* sabe tocar el *cello*, además presenta una anotación que indica que *david* lleva diez años tocando su instrumento. También podría existir un predicado sin atributos y anotaciones como *tocar()*.

Las creencias iniciales que se definen en el agente se almacenan en una *base de creencias* dinámica, ya que éstas se van actualizando de manera que nuevas creencias se añaden, otras

³Razonamiento sobre los medios para conseguir un fin

⁴En la lógica de primer orden un literal es una fórmula atómica que representa un predicado o su negación

son modificadas y otras eliminadas según la nueva información que el agente obtiene mediante diferentes medios: las percepciones que el agente obtiene del entorno (a través de mensajes con otros agentes), conclusiones razonadas por el agente (a través de las *reglas* que se detallan más adelante) o como consecuencia de acciones del agente, como por ejemplo, si su objetivo es limpiar la ventana, una vez cumplido se eliminaría la creencia de que la ventana está sucia y se añadiría la creencia de que la ventana está limpia.

Señalar que existe la posibilidad de definir reglas (que se guardan también en la base de creencias) que ayudan a razonar sobre las creencias que posee el agente para así poder deducir nuevas creencias. Las reglas están formadas por dos partes separadas por el símbolo ":-", donde a la izquierda se tiene un literal que representa la creencia que se deduce si las condiciones de la parte derecha se cumplen. Por ejemplo, la siguiente regla muestra que la creencia de que Rut es la madre de Alex se añadiría siempre que sea cierto que Rut es la madre de Daniel y a su vez que Daniel es el hermano de Alex.

```
madre_de(Rut, Alex):- madre_de(Rut, Daniel) & hermano(Daniel, Alex)
```

Objetivos

Los objetivos representan las metas que el agente desea lograr. Para ello, actuará en consecuencia hasta conseguir que se cumplan. Existen dos tipos: *achievement goals* y *test goals*. El primero, denotado por el símbolo "!" se refiere a metas que el agente ha decidido llevar a término de manera que el agente trabajará para que sean una realidad una vez alcanzados. Un ejemplo de este tipo de objetivo sería: *!llegar(Barcelona, 13:00)* que pone como objetivo que el agente llegue a Barcelona a la una de la tarde. El segundo tipo de meta, que se denota con el símbolo "?", expresa una consulta sobre la base de creencias para saber si la creencia se encuentra en la base de creencias y, en caso positivo, recoger la información de la consulta. Por ejemplo, dado el objetivo *?madre_de(Rut, X)* y la creencia *madre_de(Rut, Daniel)*, la variable "X" tomaría el valor "Daniel".

Planes

Los planes indican cómo se alcanzará la meta, es decir, qué secuencia de acciones se llevará a cabo. Constan de tres partes: el evento activador, el contexto y el cuerpo.

```
evento_activador : contexto <- cuerpo
```

Como se mencionó anteriormente, la reactividad y la proactividad son dos características propias en un agente. Así, los eventos sirven para que el agente pueda mostrar reactividad respondiendo a las percepciones que surgen del entorno, mientras de manera proactiva está actuando para conseguir su meta establecida. Cuando el evento activador se percibe como cierto, indica que el plan asociado pasaría a ser una potencial opción que se tendrá en cuenta después en el proceso de razonamiento para la elección de qué acciones ejecutar. Un evento representa un cambio en las creencias u objetivos, y ese cambio puede ser de dos tipos: una eliminación o una adición, de manera que los posibles eventos activadores quedan resumidos en la tabla 2.1.

El contexto define el "cuándo" es aplicable el plan asociado, es decir las condiciones que deben darse para que el plan pueda ejecutarse exitosamente. Está constituido por una conjunción de literales (creencias), pudiéndose usar expresiones relacionales (sintaxis similar a Prolog) y expresiones lógicas para agruparlas. Con ello el contexto de un plan es normalmente una unión

Notación	Tipo
+creencia	Adición de una creencia
-creencia	Eliminación de una creencia
+!objetivo	Adición de un <i>achievement-goal</i>
-!objetivo	Eliminación de un <i>achievement-goal</i>
+?consulta	Adición de un <i>test-goal</i>
-?consulta	Eliminación de un <i>test-goal</i>

Tabla 2.1: Tipos de eventos activadores

de los tipos de literales que se muestran en la tabla 2.2 y que todos ellos son contrastados con la base de creencias del agente de manera que es una secuencia lógica y verdadera.

Notación	Definición
l	El agente cree que l es cierto
$\sim l$	El agente cree que l es falso
not l	El agente no cree que l es cierto
not $\sim l$	El agente no cree que l es falso

Tabla 2.2: Tipos de literales en el contexto de un plan

Por último el cuerpo define el curso de acciones a ejecutar si el evento activador se unifica con un evento percibido del entorno, el contexto es cierto y el plan es elegido para ser ejecutado. Dichas acciones están separadas por ";" y hay seis tipos que pueden aparecer en el cuerpo del plan [1]:

- **Acciones externas:** representan los actos que el agente es capaz de realizar y por tanto afectar a su entorno. Por ejemplo, si el agente es un coche dispondrá de la acción *frenar()*, donde frenar es un simple predicado usado para representar simbólicamente la acción de frenar que llevarán a cabo los actuadores físicos del coche. Es por esto que hasta que no se reciba una respuesta lógica positiva por parte de los actuadores físicos, el plan que requiere dicha acción permanece pausado.
- **Acciones internas:** son acciones que se ejecutan internamente en el ciclo de razonamiento, a diferencia de las acciones externas mencionadas anteriormente que afectan al entorno y se ejecutan "fuera" del agente. Su sintaxis también es distinta, pues se representan con un punto delante del nombre de la acción, como por ejemplo *.my_name(N)*, que devuelve el nombre "N" del agente. Jason ofrece un conjunto de acciones internas estándar aunque permite también extender el lenguaje y programar nuevas acciones internas.
- **Achievement goal:** dentro del cuerpo es posible requerir que un objetivo se alcance, en cuyo caso se añade a los objetivos actuales a alcanzar que tiene el agente usando el operador "!" (*!poner_gasolina(coche)*). Este objetivo es un evento activador, como se vio anteriormente, que puede tener éxito si existe un plan para realizarlo y éste se ejecuta correctamente, o fallar en caso de que no exista un plan para dicho objetivo o se produzca algún fallo en el plan. Por ello, el plan cuyo cuerpo llame a un objetivo quedará suspendido hasta recibir la respuesta de la ejecución de dicho objetivo. Como esta situación no es deseable en muchas ocasiones, se puede usar una doble exclamación "!!" para indicar que el objetivo se añade pero el plan continúa su ejecución.

- **Test goal:** representan consultas a la base de creencias para recuperar el estado de la creencia consultada. Pueden aparecer tanto en el contexto (para comprobar si un plan es aplicable) como en el cuerpo (para usar el valor de la creencia e incluso modificarlo). Si el *test goal* falla, el plan donde aparece también fallará. Cabe destacar que para la comprobación del evento no se necesita un plan sino sólo consultar la base de creencias por lo que el coste computacional es menor que el de un *achievement goal*.
- **Notas mentales:** son creencias creadas por el agente para poder guardar y recordar más tarde información que considera relevante. Se pueden crear (*+creencia*), eliminar (*-creencia*) o modificar (*-+creencia*), donde en este último caso no haría falta la existencia previa de la creencia.
- **Expresiones:** se usan tanto en el contexto como en el cuerpo, pudiéndose emplear expresiones relacionales y aritméticas para así obtener un valor booleano. Su sintaxis es similar a Prolog, como por ejemplo: *i = .max(5, 8, 9)*, donde el operador "=" unifica "i" con el valor "9".

La sintaxis de los planes permite especificar una etiqueta que, aunque no se defina, Jason la genera automáticamente y representaría el nombre del plan. El nombre puede tener asociado anotaciones que permiten definir planes más específicos y ayudar en el proceso de selección de planes aplicables. Asimismo, gracias al nombre es posible acceder a un plan en cualquier momento de la ejecución y definir planes con el mismo nombre y distintas anotaciones, pudiendo ser unos más generales y otros más específicos. Para resumir, se muestra la sintaxis completa de un plan y un ejemplo para clarificar, recordando que las anotaciones (anot.) son opcionales.

```
@etiqueta[anot.] evento_activador[anot.]: contexto <- cuerpo
@frenar_peatones[nivel_distracción(0.5), respeto_prioridad_peatones(0.8)
+paso_de_cebra: peatones(N,M) & N≠0 & M≤4 <- !frenar_moderadamente(); -peatones(N,M);
```

Siguiendo con el ejemplo de un coche circulando por carretera, este plan llamado *frenar peatones* indica en sus anotaciones que el nivel de distracción del agente actualmente es del 50% y su predisposición a cumplir con el reglamento de dar prioridad a los peatones durante la conducción es del 80%. El evento disparador *paso de ceбра* tiene como contexto tres condiciones a comprobar: primero la existencia de la creencia *peatones* con sus correspondientes atributos, segundo que la variable "N", que indica el número de peatones cruzando, sea distinto de cero para asegurar que hay peatones cruzando y tercero que la variable "M", que indica la distancia del coche al paso de ceбра, sea menor o igual a 4 metros. Si estas condiciones se cumplen entonces el plan podría ejecutarse. Si es elegido para ejecución, se realiza la acción *frenar moderadamente* y se actualiza la base de creencias eliminando la creencia de peatones cruzando el paso de ceбра.

A continuación se presenta el ciclo de razonamiento que sigue un agente en Jason (Figura 2.3). En primer lugar, se recogen las percepciones del entorno en una lista de percepciones que llegan a la función *actualización de creencias* (AC). Esta función se encarga de plasmar en la base de creencias (BC) los cambios percibidos de manera que las nuevas percepciones que no estaban en la BC se añaden como creencias y aquellas creencias que ya no son percibidas se eliminan. Otra fuente de información que afecta a la base de creencias son los mensajes que se reciben del resto de agentes. Así, la función de *selección y aceptación de mensajes* (SAM) se encarga de revisar los mensajes recibidos, elegir uno y decidir si se acepta o no, en caso afirmativo se añade la creencia de información recibida en la base de creencias. Señalar que cuando se actualiza la BC, ya sea por los mensajes o las percepciones, se generan eventos externos que añaden o eliminan creencias. Después de que se han actualizado los eventos y la

base de creencias, la función de *selección de evento* (S_E) selecciona un evento de entre la lista de eventos actual. A continuación los planes relevantes para el evento se escogen de la biblioteca de planes (BP) y junto con las creencias actuales la función de *selección de plan* (S_P) separa aquellos que son aplicables, es decir, que tienen posibilidad de ejecutarse con éxito y de entre ellos selecciona uno. El plan seleccionado actualizará el conjunto de intenciones, y la función de *selección de intenciones* (S_I) escogerá la más apropiada. Finalmente se procede a la ejecución de la intención seleccionada y posteriormente se actualizan las intenciones, eliminándose el plan ejecutado del conjunto de intenciones. Así termina este ciclo que vuelve a repetirse.

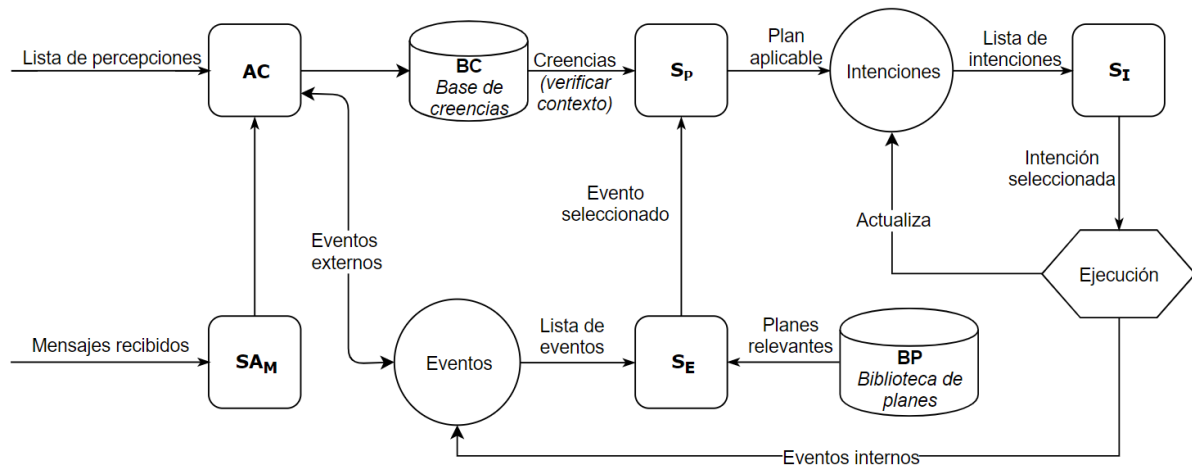


Figura 2.3: Ciclo de razonamiento de un agente en Jason

2.2 Sistemas multiagente normativos

Los seres humanos son individuos gregarios, por lo que a lo largo de la historia se han agrupado formando sociedades donde las normas en general se han empleado como mecanismo para organizar y regular la interacción entre miembros de la sociedad para así evitar conflictos y permitir una buena convivencia [15]. Con el surgir de los sistemas multiagentes, donde uno de los aspectos más relevantes recae en la autonomía que presenta el agente para actuar, los sistemas normativos pueden ser un buen instrumento de control, ya que se sirve de las normas como mecanismo para regular las acciones del agente dentro del sistema y las interacciones entre ellos. Sin embargo, existe una falta de consenso respecto a la definición del concepto de *norma*, en especial respecto a sus características y propiedades, pues han sido analizadas por distintas disciplinas y cada una ofrece una definición dentro de su campo.

2.2.1. Normas en Sistemas Multiagente

Como se describe en [16] el rol de las normas en las sociedades humanas se ha analizado desde diferentes disciplinas, como la filosofía, la sociología o el derecho. En el campo de la inteligencia artificial, las normas se han propuesto principalmente como medio para tratar problemas de coordinación en los sistemas multiagentes (MAS) [16], por medio de expresiones formales, generalmente usando la lógica deóntica [19] para formalizar las normas en términos de permisos (P), prohibiciones (F) y obligaciones (O). Con ellas se regula el comportamiento de los agentes de software y las interacciones entre ellos [20].

Jones y Carmo [5] definen un sistema normativo multiagente como un "conjunto de agentes (software o humanos) cuyas interacciones pueden considerarse reguladas por las normas, de manera que las normas prescriben cómo los agentes deberían y no deberían comportarse idealmente". Sin embargo, Boella y van der Torre [24] dieron un paso más y definieron los Sistemas Multiagente Normativos (NMAS) como "un sistema multiagente organizado mediante mecanismos para representar, comunicar, distribuir, detectar, crear, modificar y hacer cumplir las normas, deliberar sobre las normas y detectar la violación y cumplimiento de las normas". Por lo tanto, los NMAS definen normas, que son entidades inmateriales que existen gracias a su aceptación por parte de los miembros de la sociedad a fin de evitar conflictos y garantizar el orden social [18]. Además, los agentes en un NMAS deberían, al menos, ser capaces de representar estas normas y poder deliberar sobre ellas.

Varios autores han propuesto diferentes clasificaciones de normas, como las de Tuomela [22], Dignum [23], Boella [24], Savarimuthu [25] o Peng [26]. De todas estas propuestas, podemos diferenciar cuatro tipos principales de normas (ver Tabla 2.3): normas institucionales, normas sociales (convenciones), normas de interacción y normas privadas.

Tipo de norma	Promulgada por	Dirigida a	Cumplimiento	Descripción
Institucional	Autoridad institucional	Sociedad	Sanciones/Premios	SÍ (Deóntica)
Social	Surgen de las relaciones sociales	Sociedad	Mecanismos sociales (emociones)	NO
Interacción	Participantes de la interacción	Participantes de la interacción	Sanciones/Premios	SÍ (Deóntica)
Privada	Agente individual	Personal	Moral, emociones	NO

Tabla 2.3: Clasificación de las normas

Las *normas institucionales* [22, 23, 26] son promulgadas por una autoridad de la organización o la propia institución y su violación se considera un acto ilícito que conlleva sanciones o castigos, y pueden ser modeladas como leyes, como en [25]. Generalmente describen el comportamiento ideal en el sistema mediante obligaciones, prohibiciones y permisos [24], utilizando descripciones en lógica deóntica.

Las *normas sociales o convenciones* [22, 25] indican las formas establecidas y aceptadas de hacer las cosas y su violación no implica un castigo o sanción institucional pero puede haber consecuencias sociales, como ser impopular o incluso ser marginado del grupo. Tanto las normas institucionales como las sociales rigen la coordinación de los individuos en una sociedad o grupo de agentes. Sin embargo, las normas sociales no son obligadas a ser cumplidas por ninguna entidad que represente a la institución, sino que representan comportamientos que surgen de interacciones repetidas entre individuos, como una convención dentro de una sociedad que no ha sido impuesta por una autoridad central. Al ser normas emergentes, generalmente no se describen explícitamente en la sociedad, ni definen sanciones o recompensas para persuadir a los agentes de que las respeten. Pero los agentes son incitados a cumplirlas por mecanismos sociales como el ostracismo, la recriminación, etc., siendo las emociones un mecanismo interesante para forzar su cumplimiento [16].

Las *normas de interacción* [23, 26] están formadas por contratos o acuerdos formales entre entidades, que se crean explícitamente durante un período de tiempo limitado como consecuencia de una interacción entre individuos, y también se basan en la noción de obligación, prohibición y permiso, pudiéndose describir mediante la lógica deóntica. Además normalmente incluyen sanciones y recompensas.

Finalmente, las *normas privadas* [23, 26] están formadas por normas internas del agente que se auto-impone y aseguran la autonomía del agente. Estas normas privadas se crean dentro de la mente de los agentes, normalmente como resultado de la internalización de una interacción o norma social de manera que se aceptan como principios propios.

Previo a presentar las propuestas de sistemas normativos multiagente, a continuación se establece la nomenclatura que se emplea a lo largo del trabajo para la representación de las normas. Cabe destacar que la deóntica adoptada por las distintas propuestas y también por este trabajo es asumir que en el entorno todo está permitido, de manera que las normas sólo representarán las obligaciones y prohibiciones del sistema.

Definición 2.2.1. Una norma (n) se define como una tupla $n = \langle id, \{F, O\}, T, A, E, P, \{S, R\} \rangle$ donde:

- id : es el identificador de la norma.
- F, O : representa la modalidad deóntica de la norma, donde F representa la prohibición y O la obligación.
- T : es el *target* de la norma, es decir, indica a quién o quiénes va dirigida.
- A : son las condiciones de activación que determinan cuándo se activa la norma y se instancia.
- E : son las condiciones de expiración que determinan cuándo la norma expira y se cancela su efecto.
- P : representa el predicado, que puede ser un estado o acción, sobre el que la norma regula (prohíbe u obliga).
- S, R : representan la sanción o recompensa por violación o cumplimiento respectivamente, asociada a una norma.

2.2.2. Sistemas multiagente normativos

A lo largo de los años los sistemas multiagentes normativos han sido objeto de estudio debido a que ofrecen un mecanismo de control social, necesario con el auge de sistemas abiertos, distribuidos y dinámicos. La tabla 2.4 recoge las principales propuestas y presenta una comparativa de algunas características que resulta interesante destacar para así decidir el proceder de la implementación de nuestra propuesta. Se observa que la mayoría de ellas, incluidas aquellas propuestas más recientes, emplean AgentSpeak, que es un lenguaje de programación orientado a la programación de agentes que permite la creación de agentes BDI. Algunos directamente usan Jason, que es una extensión de AgentSpeak que permite, entre otras cosas, modificar el ciclo de razonamiento del agente permitiendo adaptarlo a las demandas del diseñador. Por esta razón se ha decidido centrar el estudio del estado del arte en aquellos sistemas normativos implementados con Jason.

⁵<https://github.com/meneguzzi/Iovis>

⁶<http://apapl.sourceforge.net/>

⁷<http://gti-ia.upv.es/sma/tools/magentix2/downloads.php>

⁸<http://jacamo.sourceforge.net/>

⁹<http://jason.sourceforge.net/wp/>

¹⁰<https://potassco.org/>

Sistema	Lenguaje de programación	Software descargable	Autor	Año	Referencia web
BOID	Prolog	NO	Broersen	2001	-
NoA	NoA	NO	Kollingbaum	2005	-
N-KGP	PROSOCS	NO	Sadri	2006	-
BIO	Prop. deafeasible logic	NO	Governatori	2008	-
Normative AgentSpeak	AgentSpeak	Sí(Github)	Meneguzzi	2009	5
Neto	AgentSpeak	Teórico (Jason)	Neto	2010	-
Oren	SWI-Prolog	NO	Oren	2011	-
Panagiotidi	2APL/PDDL 2.1	NO	Panagiotidi	2012	-
N-2APL	2APL	Sí (2APL)	Alechina	2012	6
MaNEA	Magentix2	Sí(Jason)	Criado	2013	7
JacaMo	AgentSpeak	Sí	Boissier	2013	8
N-Jason	AgentSpeak	Sí(Jason)	Lee	2014	-
v-BDI	AgentSpeak	Sí(Jason)	Meneguzzi	2015	9
JSAN	AgentSpeak	Sí(Jason)	Viana	2015	-
Shams	ASP	No	Shams	2017	10

Tabla 2.4: Sistemas multiagentes normativos

Se ha realizado una revisión del estado del arte de los trabajos más relevantes que emplean Jason para modelar sistemas normativos, analizando las distintas propuestas que ofrecen para: modelar o implementar las normas, el procesamiento de la norma así como el proceso deliberativo sobre la norma, entre otros aspectos. La Tabla 2.5 muestra un resumen de la comparativa de los componentes que cada autor usa para representar una norma de acuerdo con la definición de norma que se estableció anteriormente. Se observa que solamente Meneguzzi decide usar el atributo *id* para las normas. Todas las propuestas excepto [10, 12] consideran necesarias las condiciones de activación y expiración. Por su parte, las propuestas [10, 12] definen un *deadline* sobre las normas que queda plasmado en la tabla con "*". Destacar que las representaciones más completas las ofrecen [8, 9]. La tabla 2.6 muestra una comparativa de los aspectos generales de las diferentes propuestas. A continuación se describe con más detalle cada una de ellas.

Propuestas	id	Deónica(F,O)	Target	Activación	Expiración	Predicados	Sanción	Recompensa	Año
Neto[8]	-	✓	✓	✓	✓	✓	✓	✓	2010
Lee[10]	-	✓	-	*	*	✓	✓	-	2014
Viana[9]	-	✓	✓	✓	✓	✓	✓	✓	2015
Meneguzzi[11]	✓	✓	✓	✓	✓	✓	-	-	2015
Shams[12]	-	✓	-	*	*	✓	✓	-	2017

Tabla 2.5: Comparación de la representación de la norma

La propuesta de Neto et al. [8] ofrece nuevas funciones a la plataforma Jason para dar soporte al razonamiento normativo, de forma que se puedan construir agentes capaces de pro-

cesar normas, resolver conflictos entre normas y decidir libremente la acción a tomar teniendo en cuenta las normas vigentes. Su propuesta es meramente teórica y no ha llegado a ser realmente implementada en Jason, por lo que no presenta software disponible. En este caso sí se muestra el proceso del razonamiento normativo seguido, y para ello se usa una función de utilidad que representa el beneficio que se obtendrá por cumplir o violar la norma. Esta función calcula un valor que se entiende como la *contribución* (*positiva, negativa o neutra*) de la norma, la recompensa y el castigo hacia conseguir sus metas personales, i.e. si las normas le ayudan o perjudican para conseguir sus metas. Mediante el valor de esta función el agente decide que normas aceptar y cuáles rechazar, así las normas quedan separadas en dos grupos: las que se aceptan y las que se rechazan. Cada plan tiene asociada una prioridad y cuando un plan deseado por el agente además es consistente con alguna norma, la prioridad de dicho plan se aumenta. Se selecciona el plan con mayor prioridad, así el plan seleccionado también tiene en cuenta las normas y seguirá aquella que más le beneficie al agente, pudiéndose dar el caso también de violar una norma si así le conviene al agente. Con respecto a la detección y resolución de conflictos, en caso de conflicto entre dos normas, Neto propone seleccionar aquella que proporcione una mayor contribución al alcance de los deseos del agente. La propuesta consiste en modificar las funciones AC , S_E y S_P del ciclo de razonamiento Jason (ver Figura 2.4), añadiendo las siguientes funciones:

- NRF: actualiza el conjunto de normas aceptadas por el agente de acuerdo con su entorno y creencias.
- UN: actualiza las normas activas asociadas al agente, que son aquellas que en ese momento están operativas y deben respetarse.
- EN: permite decidir qué normas va a cumplir y cuáles violar. Para ello aquí se calcula la *contribución* de cada norma.
- DSC: resuelve conflictos entre normas, i.e, cuando una norma contradice a otra o cuando una norma entra en conflicto con la meta del agente.
- ADP: asigna una prioridad a los deseos y planes; a aquellos deseos que luego pasarán a ser planes que hagan cumplir objetivos y/o que respeten normas se les asigna mayor prioridad.
- SD: selecciona aquellos deseos que pasarán a ser intenciones usando las prioridades asignadas.
- SP: selecciona de todos los planes aquel con mayor prioridad.

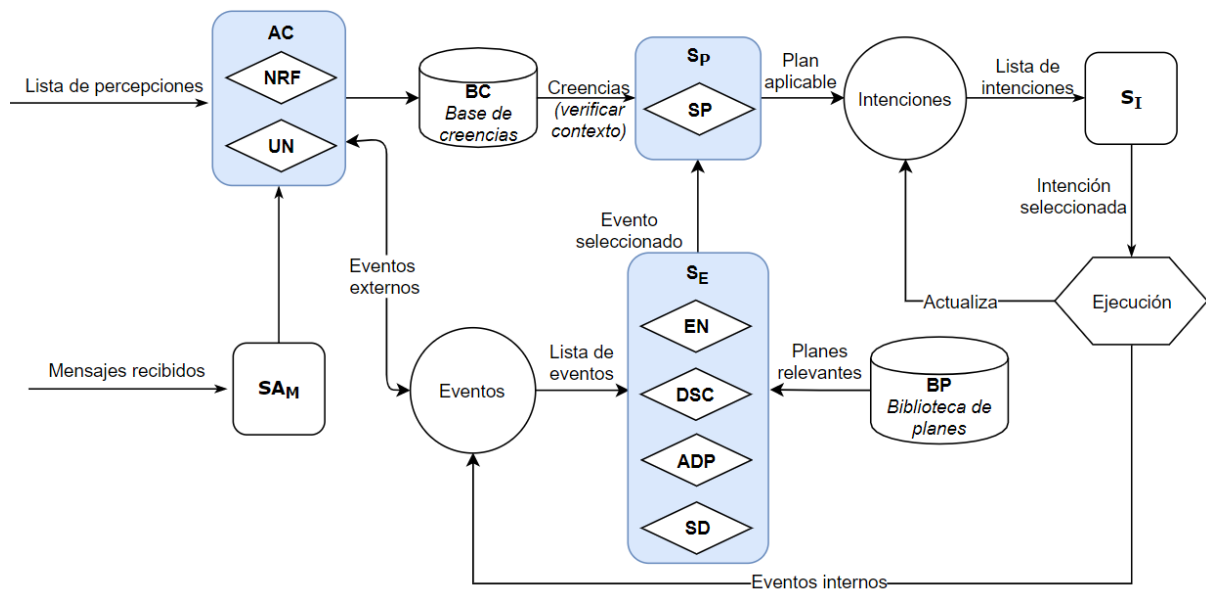


Figura 2.4: Modificaciones propuestas por Neto [8] en el ciclo de razonamiento Jason

En el trabajo de Lee et al.[10] se propone un agente normativo capaz de procesar normas y poder responder a nuevas normas que surjan. Para dar soporte a la representación de las normas, se modifica la construcción de los eventos y planes, usando las anotaciones (que son opcionales en Jason) para definir atributos como se muestra a continuación.

- Evento: se extiende la sintaxis añadiendo la especificación del tiempo de vencimiento, que indica el tiempo esperado para el cumplimiento de la intención; y la prioridad, valor numérico que expresa la importancia de responder al evento.
- Plan: se especifica la duración del plan, que luego se usará para ver si el plan es factible, i.e, si se cumpliría dentro del tiempo de vencimiento del evento que lo activa.

Además, la propuesta de Lee modifica el ciclo de razonamiento del intérprete de Jason añadiendo tres funciones para el procesamiento de las normas (ver figura 2.5):

1. *Reconsideración del evento (RE)*: distingue las normas de entre las percepciones. Verifica que la norma es ejecutable y en caso de que no tenga asociado un plan intenta averiguar cuál sería el estado o acción para hacer cumplir la norma. Una vez pasada la verificación la norma se añade a los eventos como un *achievement goal*.
2. *Reconsideración de la opción (RO)*: trata de identificar de entre los planes del agente aquel más apropiado para ejecutar la norma, es decir, determina el plan aplicable.
3. *Planificación de las intenciones (PI)*: aquí se produce el razonamiento normativo y se toma una decisión entre la meta personal del agente, las normas y las sanciones. Además, aquí se trata el conflicto entre normas que pueda haber ya que gracias a las prioridades y el tiempo de vencimiento, se ordenan las intenciones y se devuelve el conjunto de las mejores intenciones de acuerdo a estos parámetros.

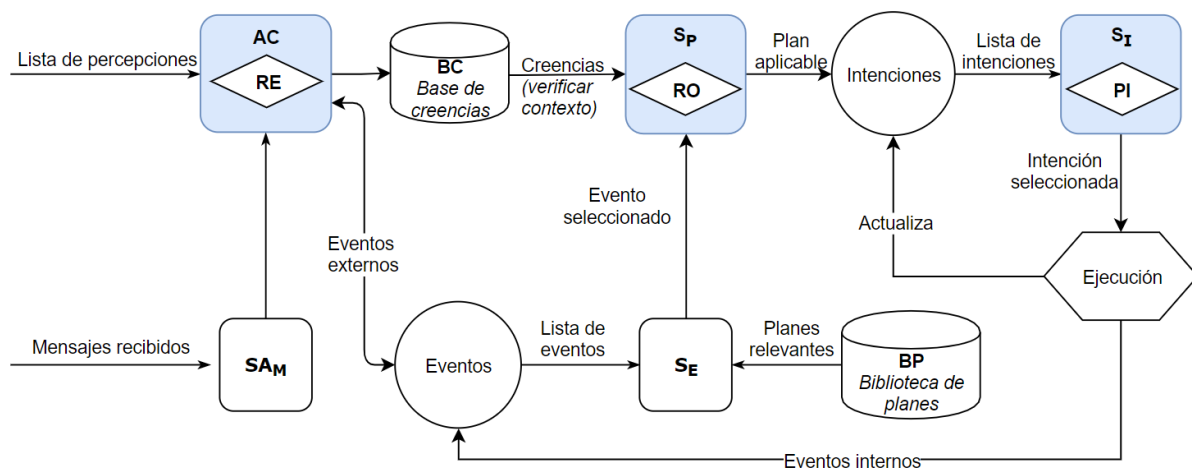


Figura 2.5: Modificaciones propuestas por Lee [10] en el ciclo de razonamiento Jason

El trabajo desarrollado por Viana et al.[9] propone la creación de un agente normativo que consta de tres componentes: *Movement Strategy*, *Generate Goal Strategy* y *Norm strategy*. Este último es el que se encarga del procesamiento y razonamiento de las normas, y consta de cuatro módulos:

1. *Norm Awareness*: donde se identifica qué normas están activas en el entorno y a quién van dirigidas (target).
2. *Norm Adoption*: el agente reconoce su responsabilidad hacia el resto de agentes interiorizando las normas que le afectan. En este punto se tiene también en cuenta la presión social.
3. *Norm Deliberation*: para decidir si cumplir o no la norma, el agente necesita saber cuáles son los objetivos que se verán entorpecidos por el castigo asociado a la norma y aquellos que se verán beneficiados (por el seguimiento de la norma y sus recompensas asociadas), para así tomar una decisión.
4. *Norm impact*: después de ejecutar la norma, se actualizan los objetivos y el ciclo del agente empieza otra vez.

Es importante destacar que en esta propuesta se tiene en cuenta un pequeño aspecto social como es la presión de grupo que se considera en el módulo *Norm Adoption* antes de ser adoptada. No se ofrece la implementación en detalle de estos módulos ni tampoco se muestra cómo estos módulos modifican el ciclo de razonamiento Jason.

En el trabajo de Meneguzzi et al.[11] se distinguen dos tipos de normas: *normas abstractas* y *normas específicas*, siendo estas últimas las únicas que se tendrán en cuenta a la hora de decidir qué plan elegir.

- Normas abstractas: el agente las percibe del entorno. Contienen la modalidad deóntica (F, O), la condición de activación (A), la expiración (E) y el identificador de la norma (id).
- Normas específicas: son normas abstractas instanciadas dada una base de creencias, es decir, las normas que se encuentran activas. Una norma específica representa una sustitución que satisface todas las condiciones de activación de la norma abstracta y contiene la

acción/plan a realizar en respuesta, la condición de expiración y un identificador único de instanciación.

Una norma abstracta puede ser activada múltiples veces con la misma sustitución pero con distinto identificador. En la figura 2.6 se muestra el ciclo de razonamiento Jason actualizado según la propuesta de Meneguzzi. Cada vez que la base de creencias se actualiza, también lo hace el conjunto de normas específicas (AN) y se resuelven los conflictos que surjan (RC). El proceso de razonamiento de la norma viene en la etapa de selección del plan (SP). Así, dado un plan que sea factible se comprueba si viola o no alguna norma específica, de manera que para seleccionar el plan, de todos aquellos planes factibles, se usa una función de utilidad $f : baseUtility(plan) + fulfilUtility(plan) - violCost(plan)$ y se escoge aquel plan que maximice esta función. El parámetro *baseUtility* cuantifica el beneficio que proporciona el plan, si decidir por ese plan lo acerca o aleja de sus objetivos; *fulfilUtility* es el beneficio que proporciona cumplir con las normas que se encuentran en el plan; y *violCost* representa la suma de las sanciones de las normas violadas. En caso de conflicto, la resolución que se adopta es seleccionar aquel plan que presenta una mayor función de utilidad.

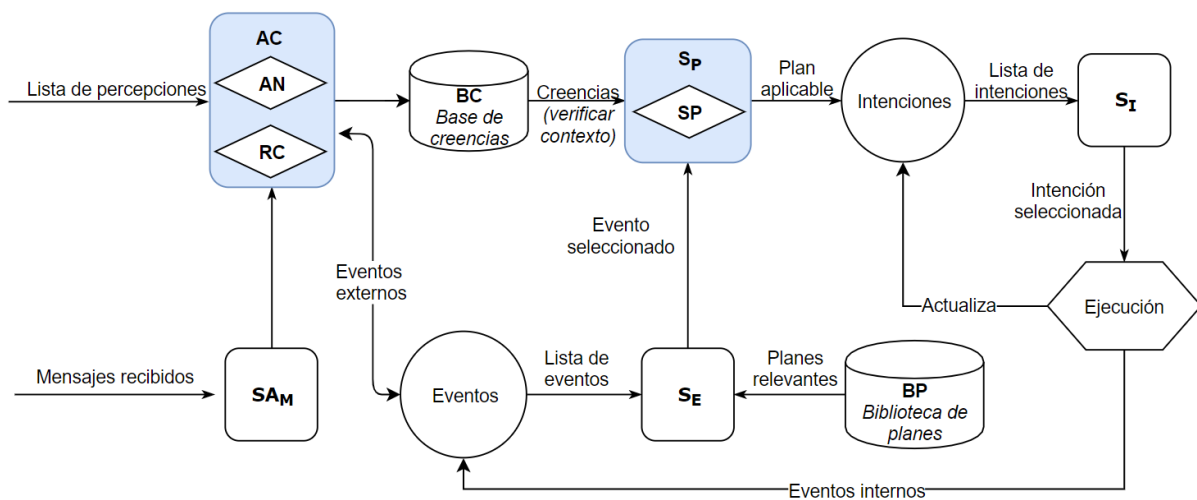


Figura 2.6: Modificaciones propuestas por Meneguzzi [11] en el ciclo de razonamiento Jason

Por último, mostramos la propuesta de Shams et al.[12], que a pesar de no emplear Jason como lenguaje de programación, se ha decidido comentar su trabajo por ser la propuesta más reciente sobre la implementación de un sistema multiagente normativo. Los autores se centran más en la implementación del razonamiento normativo (considerando las normas durante la generación y selección del plan) y además soporta planes con más de una meta. Las normas, como en la propuesta anterior, pueden ser instanciadas varias veces; las acciones son durativas y también se define un *deadline* asociado a las normas. Cada plan tiene asociado una función de utilidad que se calcula como el beneficio por alcanzar dicho objetivo menos el coste de violar normas asociadas a ese plan; $f : v(g) - c(n)$. Así se calcula el conjunto de planes óptimos y se elige de entre ellos aquel que maximice la función de utilidad.

Para concluir este análisis, cabe destacar que de las distintas propuestas vistas, la representación normativa de Meneguzzi[7] no especifica la sanción o recompensa asociado a cada norma, de manera que el agente ignora esta información y por tanto tampoco la puede usar en el proceso de razonamiento, como mecanismo para hacer respetar las normas. Asimismo, en la propuesta de Viana [9] tampoco se detalla el proceso de razonamiento normativo ni el pro-

ceso de selección del plan. Sin embargo, cabe destacar que sí toma en cuenta la representación de las recompensas y las sanciones en las normas, y además es el único que tiene en cuenta un pequeño aspecto social como es la presión social. Por otro lado, la propuesta de Neto [8] resulta interesante, aunque se queda solo en un marco teórico y no hay código que indique cómo se implementarían las funciones propuestas. Finalmente, en la propuesta de Lee [10] no se expone de manera clara la representación de la norma que usa, aunque cabe destacar que en lugar de usar condiciones de activación y desactivación, usa el atributo *deadline* que básicamente establece lo mismo, ya que la activación y desactivación de la norma se modela como un periodo temporal establecido por el *deadline*.

	Implementación	BDI	MAS	Resolución de conflictos	Modelización de la norma	Razonamiento normativo	Selección de plan	Software asociado	Caso de estudio	Año
Neto[8]	-	✓	-	✓	percepción	✓	F. utilidad	✓	Operación de rescate	2010
Lee J.[10]	AgentSpeak	✓	-	✓	percepción	-	Prioridades	✓	Robot trae cerveza	2014
Viana[9]	AgentSpeak	✓	✓	-	percepción	✓	F. utilidad	-	Evacuación de áreas de riesgo	2015
Meneguzzi[11]	AgentSpeak	✓	✓	✓	percepción	✓	F. utilidad	✓	Emergencia por inundación	2015
Shams[12]	ASP ¹¹	-	-	✓	-	✓	F. utilidad	-	Recuperación de desastres	2017

Tabla 2.6: Comparativa de los aspectos generales de cada propuesta

2.3 Sistemas multiagente normativos emocionales

En el campo de la Inteligencia Artificial, para dar más realismo a los agentes e interactuar con ellos de una manera más creíble, el pensamiento y los procesos de decisión de los agentes deben parecerse a la forma humana de pensar y tomar decisiones. Por esta razón, los agentes deberían tener en cuenta, por un lado, las normas del contexto en el que se encuentran (es decir, las normas sociales, organizativas, internas, etc.), y por otro, las implicaciones y consecuencias emocionales que puede generar el cumplimiento o incumplimiento de estas normas.

Por ejemplo, consideremos un escenario virtual donde hay un conjunto de agentes dentro de una organización de taxistas. La organización ha establecido un conjunto de normas y reglamentos. Una de estas reglas implica que los taxistas tienen que formar una cola para recoger a los clientes en las estaciones de taxi, por lo que el último conductor que llega debe ir al final de la cola. Si un taxista incumple esta norma, y cuando llega a una estación va al principio de la cola para ser el primero en recoger a los clientes, el agente podría haber logrado su objetivo de obtener la mayor cantidad de clientes por día. Sin embargo, la violación de la regla puede generar emociones negativas. Por un lado, los otros agentes pueden generar emociones, como

¹¹ Answer Set Programming

ira o reproche, que podrían desencadenar otras acciones en el entorno (por ejemplo, aislamiento social del agente que viola la norma). Por otro lado, el conductor que violó la norma podría generar, dependiendo de su personalidad, emociones de satisfacción (por lograr sus objetivos) o vergüenza (por haber sido sancionado por sus colegas). Además de eso, la próxima vez que este agente se enfrente a una situación similar, podría considerar las consecuencias emocionales de sus acciones anteriores para tomar decisiones futuras.

Por tanto, la combinación de modelos emocionales y modelos normativos parece interesante para así modelar agentes capaces de tratar las normas del grupo y también las emociones que surgen por el uso de las normas, sus consecuencias e interacciones con otros agentes.

2.3.1. Emociones en Sistemas Multiagentes

Una emoción es un estado afectivo de conciencia en el que se experimenta alegría, tristeza, miedo, odio o similares, a diferencia de los estados de conciencia cognitivos y voluntarios ¹².

El campo de la computación afectiva tiene como objetivo integrar emociones en sistemas inteligentes para dotarlos de capacidades para reconocer, sentir, inferir e interpretar emociones humanas. Las emociones desempeñan un papel clave en muchos procesos, como la comunicación, la toma de decisiones, el aprendizaje, etc. Por lo tanto, es importante intentar proporcionar a los agentes modelos computacionales de emociones para proporcionarles capacidades emocionales[38].

Los *modelos computacionales de emociones* (CME ¹³) [28] son sistemas de software diseñados para proporcionar agentes autónomos con mecanismos adecuados para el procesamiento de información emocional, la obtención de emociones sintéticas y la generación de comportamiento emocional. Su desarrollo está muy influenciado por teorías psicológicas y modelos de emoción. En general, las CME incluyen mecanismos para la evaluación de estímulos emocionales, la obtención de emociones y la generación de respuestas emocionales. Los CME permiten dotar a los agentes con habilidades para el reconocimiento de emociones de personas y de agentes artificiales, la simulación y expresión de sentimientos emocionales, y la ejecución de respuestas emocionales. Los enfoques teóricos más notables que han influido en el desarrollo de CMES son: (i) teorías de valoración, (ii) las teorías dimensionales y (iii) las teorías jerárquicas de la emoción.

Las *teorías de valoración* diferencian las emociones de acuerdo a la relación entre los individuos y su entorno [29, 30, 31], de modo que las emociones surgen de la evaluación de situaciones, objetos y agentes existentes en el entorno, que directa o indirectamente afectan a las metas, los planes y las creencias del individuo [28]. Se han propuesto varias instancias de esta teoría, siendo el modelo de evaluación de OCC [29] de Ortony et al. uno de los más implementados en CMEs [32, 42, 43]. Este modelo considera 22 emociones diferentes como reacciones positivas o negativas provocadas por los aspectos de *objetos* (amor, odio), las *acciones* que realizan los agentes (orgullo, admiración, vergüenza, reproche), y los efectos de los *eventos* generados por la acción de otros agentes (feliz por, decepción, esperanza, satisfacción, alivio, alegría, miedo, lástima, angustia, resentimiento, regodearse, miedos confirmados). Además, estas emociones se combinan para dar lugar a un conjunto de emociones compuestas, es decir, emociones relacionadas con los efectos de los eventos causados por las acciones de los agentes (gratificación, ira, remordimiento, gratitud).

¹²WordReference Rando House Unabridge Dictionary of American English, 2017

¹³Computational models of emotions

Otra *teoría de valoración* notable es el modelo que propone Frijda [30], que se centra en el proceso de la emoción, definiendo tres fases: (i) la evaluación de la emoción desde el entorno; (ii) el impulso o instigación de la tendencia de una acción; y (iii) la generación de acciones cognitivas, posiblemente en forma de comportamientos principalmente expresivos, como por ejemplo expresiones faciales. Además existen CMEs basados en las teorías de valoración como por ejemplo EMotion and Adaptation (EMA) [44].

Las *teorías dimensionales* sobre la emoción representan las emociones desde una perspectiva estructural, estableciendo lo que se puede diferenciar tomando como base parámetros dimensionales, como "incitación" y "valencia" [28]. Ejemplos relevantes de esta teoría son la propuesta de un marco bidimensional de Russell [45], que se caracteriza por el agrado/desagrado y activación/desactivación de una variedad de fenómenos afectivos como emociones, estado de ánimo y sentimientos; y la propuesta de un marco tridimensional de Russell y Mehrabian [46], también conocido como modelo PAD¹⁴, que describe las emociones en función del nivel de agrado, activación y dominio. Este modelo PAD también se ha utilizado para representar escalas de temperamento y describir tipos de personalidad [47]. Ejemplos de CMEs basadas en teorías dimensionales son WASABI¹⁵ (simulación de afecto para agentes con interactividad creíble) [48], Alma¹⁶ (un modelo de capas de afecto) [49] y GENIA3¹⁷ (una arquitectura afectiva inteligente de propósito general) [51]. Esta última propuesta también proporciona una extensión a Jason [52], el lenguaje de referencia para los agentes de BDI. Esta extensión permite definir los rasgos de personalidad de los agentes, la racionalidad, las estrategias de afrontamiento y las diferentes categorías afectivas, según el modelo PAD.

Finalmente, en las *teorías jerárquicas* de las emociones, existe un pequeño conjunto de emociones básicas, primarias o fundamentales, que tienen una base evolutiva y son innatas e instintivas, y han sido ampliamente investigadas e identificadas. Las emociones básicas se consideran como bloques de construcción que permiten la construcción de emociones más complejas. El grupo más aceptado de emociones básicas fue establecido por Ekman [54], su propuesta consta de seis emociones: ira, sorpresa, felicidad, aversión, tristeza y miedo, que son comunes alrededor del mundo a pesar de las diferencias culturales. Ejemplos de CMEs basadas en teorías jerárquicas son Cathexis [55] y WASABI (que combina teorías dimensionales con teorías jerárquicas).

2.3.2. Relación entre normas y emociones

En este trabajo, hemos considerado cuatro tipos diferentes de relaciones entre las normas y las emociones. La Figura 2.7 ilustra los cuatro tipos de relaciones o conexiones que se han encontrado entre las emociones y las normas: (1) las emociones pueden afectar directamente el proceso de decisión de cumplimiento de la norma, de manera que el agente tiene en cuenta las emociones (emociones actuales y previas) para decidir si cumplir o no con una norma específica; (2) el cumplimiento o transgresión de una norma puede generar emociones (positivas o negativas) en el agente que realiza la acción controlada por la norma, o en los observadores de dicha acción. También la sola existencia de la norma condiciona las emociones, de manera que el agente adecúa sus emociones para reaccionar como la situación lo demanda, como en el caso de las normas sociales (por ejemplo en un funeral las emociones alegres quedarían relegadas ante aquellas que muestren tristeza); (3) las emociones pueden usarse como un mecanismo pa-

¹⁴ Pleasure, Arousal and Dominance

¹⁵ Affect Simulation for Believable Interactivity

¹⁶ A Layered Model of Affect

¹⁷ a General Purpose Intelligent Affective Architecture

ra hacer cumplir las normas sociales, para que el agente tenga en cuenta sus propias emociones o las generadas por el resto de agentes respecto al cumplimiento o transgresión de una determinada norma, y así el agente queda condicionado a la hora de tomar su decisión; y (4) las emociones pueden ayudar a la inferencia de normas privadas, así algunos comportamientos generalmente aceptados son reafirmados de esta forma.

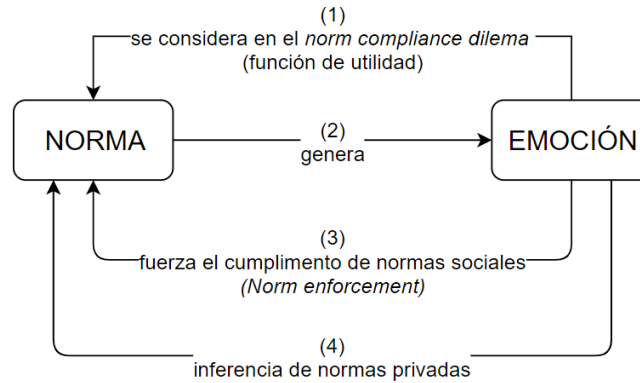


Figura 2.7: Relación normas y emociones

Respecto a la primera relación, hemos considerado el uso de las emociones como parámetro dentro del razonamiento normativo, conocido también como *norm compliance dilemma* [16]. Así, tomar una decisión sobre transgredir (o cumplir con) una norma debe considerar la utilidad esperada de esta decisión en términos del efecto sobre los objetivos del agente, la coherencia de esta decisión con respecto a las cogniciones del agente y las consecuencias emocionales de estas decisiones. Como Criado et al. [16] argumentan, las decisiones sobre si cumplir o no con una norma no sólo tienen que basarse en decisiones racionales (como normalmente ocurre) sino que también se deben tener en cuenta las emociones para proporcionar un sistema más realista y una solución compleja al problema de la toma de decisiones.

Esta apreciación de las emociones también está ligada al concepto de la moralidad. Así, a medida que los agentes se vuelven más autónomos, pueden tener que enfrentar dilemas morales en sus procesos de toma de decisiones [40]. No todas las decisiones se pueden tomar en base a marcos basados en la utilidad. Existe un área de investigación dentro de la Inteligencia Artificial, que estudia cómo la moralidad influye en el proceso de toma de decisiones de ciertas acciones [71, 72]. Por ejemplo, Conitzer et al. [41] analizaron cómo los efectos morales pueden incluirse en el proceso de decisión utilizando la teoría de juegos o los enfoques de aprendizaje automático. Shaw et al. [71] propuso un agente inteligente artificial que incorpora el aprendizaje de una "máquina moral" para su proceso de toma de decisiones. Malle et al. [74] realizó una serie de experimentos para analizar cómo aplicar las normas morales a los agentes robóticos y hacer juicios morales sobre su comportamiento.

Por otro lado, Joffily et al. [59] analizan la relación entre emociones, sanciones y cooperación, donde cooperación puede verse como la decisión de seguir las reglas. Su trabajo muestra que las emociones desencadenadas (en otros agentes y en sí mismo) como consecuencia de la decisión tomada por el agente afectarán a su comportamiento para que el agente esté más dispuesto a seguir la norma. De modo que las emociones influyen en el agente "obligándolo" a seguir la norma, en el comportamiento del agente y en última instancia en sus relaciones sociales.

Respecto a la segunda relación que se establece, i.e. la acción de respetar o infringir la norma puede provocar emociones negativas o positivas tanto en el agente mismo, como en los agentes que observan la acción. De este modo, por ejemplo tomando el modelo OCC, la

acción del agente a seguir la norma daría lugar a emociones como la satisfacción o la alegría y gratificación o admiración en los agentes que observan la acción; mientras que la transgresión de la norma evocaría emociones como miedo o resentimiento en el agente y reproche o ira en los agentes que observan la acción.

En relación con la tercera conexión establecida, las emociones también pueden ser usadas como mecanismo para hacer cumplir normas sociales [16]. Por ejemplo la violación de una norma social puede desencadenar emociones negativas como vergüenza o culpa en el transgresor de la norma, incluso si nadie notó que la norma haya sido violada [56]. De este modo, las emociones pueden surgir como consecuencias internas negativas de infringir la norma social (o positivas por respetar la norma social) y por lo tanto pueden servir como mecanismos para aplicar o mantener normas sociales, además del uso de sanciones o recompensas externas. Así mismo, estas emociones resultantes también pueden diferir dependiendo de quién infringe la norma social. Por ejemplo, la vergüenza y culpabilidad son emociones que surgen cuando el mismo agente vulnera una norma, mientras que el desprecio y la ira están supeditados a una violación de la norma por parte de otro/a. Staller y Petta [56] afirman que las personas se sentirán avergonzadas si son aisladas por no seguir las normas sociales. Por ejemplo, cuando alguien viste con vaqueros en una cena formal donde todos los demás van vestidos correctamente, la persona en cuestión podría sentirse avergonzado/a por no haber seguido la norma social. Si el resto del grupo de alguna manera muestra desdén hacia la persona transgresora, se refleja así cómo las emociones se usan como una forma de sancionar las transgresiones de las normas sociales [56].

Finalmente, las emociones pueden servir para inferir o interiorizar normas. Así, existen trabajos en ciencias sociales que sostienen que la anticipación de las emociones promueve la internalización y el cumplimiento de las normas [60]. Por ejemplo, el trabajo descrito en [61] modela la ejecución de normas sociales en sociedades de agentes basadas en emociones. En este enfoque, la sociedad supervisa el cumplimiento de las normas y genera emociones sociales como el desprecio o el disgusto en el caso de violación de la norma, y admiración o agradecimiento en el caso del cumplimiento de la norma. De manera similar, los agentes observan la expresión de estas emociones y son capaces de generar emociones como la vergüenza o la satisfacción en respuesta. Las asociaciones entre la emoción negativa y la transgresión de la norma “fomentan la (re) internalización y refuerzo de las normas sociales y, al mismo tiempo, actualizan la correspondiente representación interna de la norma”. Del mismo modo, “las situaciones de cumplimiento de normas se asocian e internalizan junto con las emociones positivas, lo que motiva a un actor a buscar situaciones en las que el cumplimiento con la representación interna (actualizada) de una norma conduce a la recompensa de emociones positivas intrínsecamente”.

Ahora bien, todos estos tipos de relaciones entre normas y emociones se dan fundamentalmente en entornos virtuales, donde resulta crucial que los agentes virtuales muestren como respuesta las reacciones emocionales asociadas al cumplimiento o violación de las normas [57], para aumentar la credibilidad (o realismo) de estos agentes. Para que los agentes inteligentes sean creíbles, deben respetar y seguir las reglas sociales (normas y/o convenciones institucionales) establecidas dentro del entorno social virtual. Las reacciones emocionales de los agentes no solo resultan del hecho de que sus metas están satisfechas, sino también de las acciones realizadas en el entorno social, como la violación de una norma social importante, incluso si esa acción contribuyó al éxito de un objetivo personal del agente. [57].

2.3.3. Sistemas normativos emocionales

La tabla 2.7 muestra una comparación de los trabajos más relevantes que han considerado la relación entre las emociones y las normas. La primera columna contiene el primer autor del artículo donde se describe la propuesta, así como su referencia bibliográfica. La segunda columna es el año de la publicación. Las dos columnas siguientes describen la teoría emocional utilizada en la propuesta y la arquitectura emocional empleada. La quinta columna indica la arquitectura del agente utilizada para el sistema multiagente. La sexta columna indica el formalismo utilizado para la representación de la norma. La séptima columna muestra qué relación entre normas y emociones se ha tenido en cuenta. La octava columna describe el escenario donde se aplicó la propuesta. Y finalmente, la última columna indica el nombre de la arquitectura final propuesta.

Propuestas	Año	Teoría Emocional	Arquit. emocional	Arquit. del agente	Represent. de las normas	Rel. Normas /Emoción	Escenario	Arquit. final
Staller[56]	2001	Frijda	TABASCO	JAM(BDI)	Reglas If-then-else	(1)Norm compliance	Control de agresión	TABASCO _{JAM}
Bazzan[64]	2003	OCC	OCC traducido a sistema basado en reglas	No especificado	Reglas If-then-else	(1)Norm compliance	Control de agresión	
von Schevel[65]	2006	Elster's	MULAN	SONAR	Petri Nets	(1)Norm compliance (2)Norm enforcement (3)Inferencia de la norma		
Ahmad[66]	2012	OCC	OCC traducido a sistema basado en reglas	OP-RND (BDI)	Metas normativas	(4)Generación de emociones	EPMP	OP-NRD-E
Ferreira[57]	2014	OCC	FAtiMA	FAtiMA (BDI)	Lógica deóntica	(4)Generación de emociones	Fumar	

Tabla 2.7: Estado del arte de Sistemas Normativos Emocionales

Staller y Petta [56] propusieron TABASCO_{JAM}, que es una arquitectura basada en agentes que combina la arquitectura TABASCO¹⁸ agente-afectiva [62] y una arquitectura BDI, llamada JAM [63]. La arquitectura TABASCO_{JAM} capta los principales componentes del proceso de emoción (evaluación, impulso y acciones cognitivas), detallados por la teoría de evaluación de Fridja [30]. Sus principales etapas son: (i) el componente "Observador" detecta el mundo y actualiza el "Modelo del mundo" (una base de datos que representa las creencias del agente); (ii) el componente "Evaluación" evalúa las creencias del "Modelo del mundo" con respecto al resultado de la evaluación y calcula un valor de intensidad, que después (iii) el componente "Impulso" utiliza para publicar un objetivo en la "Estructura de intención"; (iv) los planes en la "Biblioteca de Planes" aplicables a las metas publicadas por el componente "Impulso" contienen las acciones que se ejecutarán; y (v) un proceso regulatorio en el componente de "Evaluación" determina si la ejecución de una instancia del plan da como resultado la violación de una norma, y el plan de meta nivel utiliza el resultado de la evaluación y el valor de intensidad para determinar si se obedece o infringe la norma. Por lo tanto, las normas sociales se implementan aquí como una regulación del comportamiento general, por medio de reglas *If-then-else* cableadas en los agentes. Los autores emplearon el caso de estudio "Control de agresión" [69] para validar su propuesta. Podemos afirmar que su propuesta se centra en el uso de normas para el dilema de cumplimiento de normas (ver Figura 2.8), sin embargo la deliberación normativa

¹⁸Tractable Appraisal-Based Architecture for Situated Cognizers

o el razonamiento normativo no se ha tenido en cuenta, por lo que la arquitectura JAM no se puede considerar claramente como una arquitectura normativa.

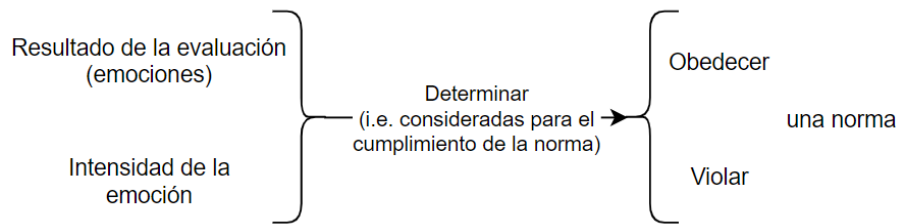


Figura 2.8: Emociones utilizadas para el dilema de cumplimiento de las normas en Staller

Bazzan et al. [64] definen un marco que permite a los usuarios definir las características de una interacción dada, las emociones que los agentes pueden mostrar y cómo estas emociones afectan sus acciones e interacciones. Las normas se relacionan aquí con las interacciones que los agentes siguen cuando se encuentran. Respecto a la parte emocional de su marco, traducen el modelo OCC en un sistema basado en reglas que genera emociones cognitivas en un agente. Estas reglas de *If-then-else* prueban la deseabilidad (de una consecuencia de un evento), la bondad (de una acción de un agente) o el atractivo (de un objeto). La regla determina el potencial para generar un estado emocional en consecuencia. Además, de forma similar al trabajo de Staller y Petta, las normas sociales se implementan aquí directamente en los agentes como parte de estas reglas *If-then-else*.

Von Scheve et al. [65, 61] basaron su enfoque en el análisis de las emociones de Elster, quien determinó que las emociones son el resultado de mecanismos y pueden desencadenar otros mecanismos[60]. Elster sostiene que imponer sanciones al infractor de la norma es impulsado por emociones como el desprecio, el desdén o la indignación, que conllevan emociones negativas (como por ejemplo: remordimiento, culpa o vergüenza) en el infractor. Von Scheve et al. propone un modelo basado en redes de Petri¹⁹ que combina SONAR (una arquitectura de agentes sociológica) y MULAN (una arquitectura multiagente) para modelar entidades sociales formadas por diferentes capas. Así, usan MULAN para implementar conceptos clave como autonomía, movilidad, cooperación y adaptación; y la arquitectura SONAR para modelar las representaciones internas de una entidad (reconocimiento, observación y acciones). En su enfoque, los agentes pueden observar el comportamiento de los demás y percibir la transgresión de la norma. De este modo, se provocan emociones sociales de desprecio, desdén o repugnancia y su expresión constituye la sanción de un infractor normativo que genera emociones negativas en el violador e induce estados de remordimiento, culpa o vergüenza (ver Figura 2.9). Aunque se trata con normas sociales, esta propuesta carece de una representación explícita de las normas (por ejemplo, mediante el uso de la lógica deóntica); y su modelado en Redes de Petri implica redes de referencia (con redes recursiva) que podrían hacer que el modelado de las emociones y las normas sea bastante complejo. Su propuesta se centra en el cumplimiento de las normas, la aplicación de normas y la inferencia de normas privadas (ver figura 2.9).

¹⁹Representación matemática o gráfica de un sistema a eventos discretos en el cual se puede describir la topología de un sistema distribuido, paralelo o concurrente (Wikipedia)

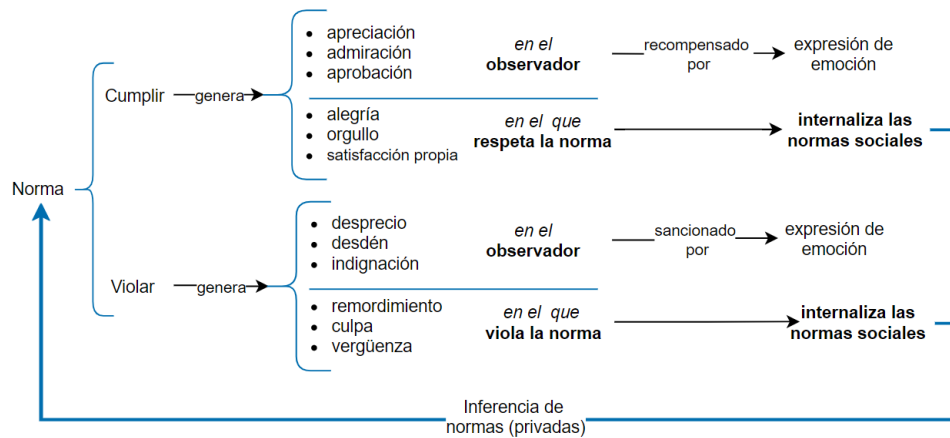


Figura 2.9: Relación de las normas y emociones en la propuesta de Von Scheve et al.

Ahmad et al. [66] proponen el marco OP-RND-E, basado en el marco normativo OP-RND [67], en el que las normas se modelan como obligaciones ante la autoridad para realizar una acción específica en una situación de tiempo limitado. Su modelo emocional se basa en la teoría de OCC, de la cual solo consideran emociones de "alegría", "orgullo", "angustia" y "vergüenza" para representar las categorías positivas y negativas de las emociones, respectivamente. Los eventos están representados por la ocurrencia de metas, que pueden ser objetivos normativos, metas personales obligatorias y objetivos personales discrecionales. Por lo tanto, las normas se modelan aquí como objetivos normativos, que representan las acciones que deben realizarse dentro de un tiempo estipulado. Las emociones se activan cuando ocurren eventos inesperados, y el agente necesita usar sus recursos y esfuerzos para completar las tareas para lograr el objetivo normativo. Por ejemplo, una emoción positiva (alegría) se desencadena al obtener tiempo adicional para lograr el objetivo normativo y la emoción positiva de "orgullo" por la capacidad de realizar la acción en el tiempo establecido, mientras que una emoción negativa (angustia) se desencadena por perder tiempo para lograr el objetivo normativo y "vergüenza" por la incapacidad de realizar la acción en el tiempo. Por lo tanto, los cambios de los eventos determinan las emociones positivas o negativas que influyen en la deseabilidad del agente. Si la emoción provocada es negativa, el agente debe volver a evaluar sus planes. Por lo tanto, las emociones motivan a un agente a planificar mejores acciones para alcanzar su objetivo normativo. Como resultado, podemos afirmar que su propuesta se centra principalmente en la generación de emociones y, de alguna manera, en el cumplimiento de las normas (ver figura 2.10). Los autores consideraron el caso de estudio del "Proceso de preparación de un examen y el proceso de moderación" (EPMP) para validar su propuesta.

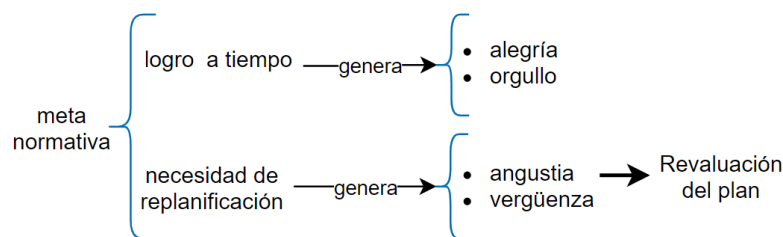


Figura 2.10: Generación de emociones de acuerdo a la propuesta de Ahmad et al.

Finalmente, Ferreira et al. [57, 58] se centraron en cómo aumentar la credibilidad de los agentes con la representación de personajes virtuales al generar emociones no solo a partir de

los eventos que afectan a las metas de un personaje, sino también de otras fuentes de emociones, como las normas y los estándares. Por lo tanto, propusieron un modelo para la generación de emociones basado en la evaluación de acciones asociadas con eventos relacionados con la norma, como el cumplimiento o la violación de una norma. Su modelo incluye aspectos sociales como el contexto normativo, las relaciones dentro y fuera del grupo, los roles sociales y los comportamientos socialmente aceptables descritos por las normas sociales que están activas en un contexto dado para inferir el estado emocional. Las emociones son el resultado de la evaluación de acciones que cumplen o quebrantan las normas sociales. Las emociones (i.e., orgullo, vergüenza, admiración y reproche) se activan en función de las variables evaluadas: elogiable/reprochable (es decir, qué tan socialmente aceptable o reprehensible era esa acción), expectación-desviación (es decir, cuán inesperada fue la acción) y unidad de fuerza cognitiva (es decir, similitud entre los atributos de los usuarios y la relación social). Los autores consideraron como caso de estudio un escenario virtual en el que los agentes fumaban dentro de un bar, para validar su propuesta. Para ello utilizaron la arquitectura de agente FATiMA²⁰ [68], una arquitectura BDI que otorga a los agentes la capacidad de generar reacciones emocionales ante los eventos, basados en el modelo OCC pero en los que no había una noción explícita de normas. Por lo tanto, complementaron esta arquitectura con un modelo normativo, en el que las normas incluyen: condiciones de activación, condiciones de expiración, condiciones normativas (prescripciones para el comportamiento de los agentes), objetivos (agentes que se espera cumplan la norma) y relevancia (importancia de la norma). Además, su modelo emocional desencadena "Emociones de Atribución", i.e. "orgullo" y "remordimiento", que ocurren cuando el agente evalúa sus propias acciones como loables (al cumplir una norma) o reprochables (cuando viola una norma), respectivamente, así como emociones de "admiración" y "reproche", que surgen de la valoración de acciones de otros como loables o reprochables. En su propuesta, los agentes comprueban constantemente si alguna norma se activa o expira. Cada vez que un agente percibe un nuevo evento, comprobará si se trata de una acción de un agente que causa el cumplimiento o la violación de una norma. Cuando se detecta un cumplimiento de norma, el agente evalúa ese evento y calcula cuán loable es y su expectación-desviación para determinar la intensidad de la emoción resultante. Así, esta propuesta se centra principalmente en la generación de la emoción (ver Figura 2.11)

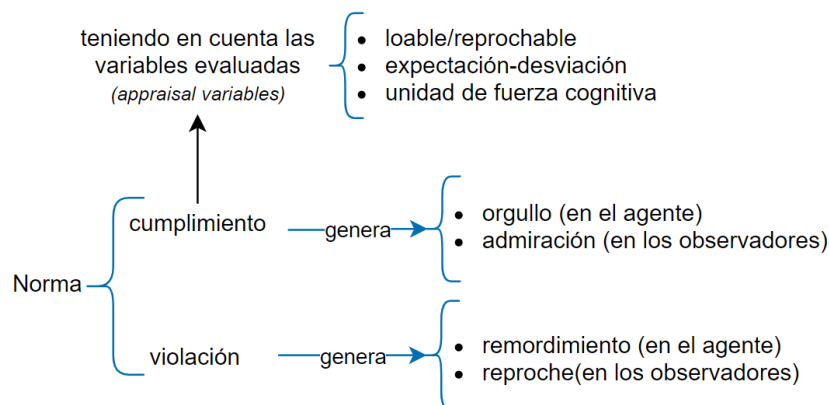


Figura 2.11: Generación de emociones de acuerdo a la propuesta de Ferreira et al.

²⁰Fearnote AffecTive Mind Architecture

2.4 Discusión

En resumen, en este capítulo se ha presentado el trabajo más relevante sobre los sistemas multiagente normativos (en Jason) y los sistemas normativos emocionales. El análisis realizado muestra la utilidad de los sistemas multiagente normativos para controlar el comportamiento e interacción de los agentes en entornos abiertos y heterogéneos (como son los sistemas distribuidos cada vez más en auge). Se ha visto que conviene modelar las normas de manera que el agente pueda razonar sobre ellas, para así respetar la autonomía del agente a poder decidir si violar o cumplir con las normas. Ahora bien, también se ha puesto de manifiesto el papel que juegan las emociones en el proceso de razonamiento práctico y por ende en el proceso de *razonamiento normativo* o *dilema del cumplimiento de la norma*. Por esta razón el interés en el desarrollo de sistemas multiagente normativo-emocionales donde los agentes sean capaces de tratar con las normas así como con las emociones.

Del análisis realizado, se observa que la creación de agentes normativos emocionales es un campo muy reciente y aún por explorar, ya que ninguna de las propuestas analizadas da total soporte a las cuatro relaciones que se encontraron entre las emociones y las normas. Algunos aspectos que cabe comentar después del análisis del estado del arte son:

- *Dilema del cumplimiento de la norma*: los agentes normativos deben estar dotados de capacidades para reconocer, representar y aceptar normas, y para resolver posibles conflictos entre ellos. Como Criado et al. argumentan [16], las decisiones sobre si cumplir o no una norma no solo tienen que basarse en decisiones racionales, sino que también deben tomarse en cuenta las emociones para proporcionar una solución más realista y compleja al problema de la toma de decisiones. Actualmente, los trabajos que se han centrado en este problema han implementado principalmente normas que utilizan reglas *if-then-else* codificadas en los agentes. Sin embargo, todavía hay una necesidad de mecanismos que hagan uso de una representación explícita de las normas, así como una representación explícita de las emociones, para considerar fenómenos como la vergüenza, el honor, la gratitud, etc. en los procesos de toma de decisiones de cualquier tipo de normas (normas institucionales, convenciones, normas de interacción y normas privadas). Las emociones que deben tomarse en cuenta deben ser no solo las emociones actuales evaluadas de eventos previos del entorno, sino también las *emociones anticipadas* [14] que se corresponden con las emociones que se dispararían en caso de infringir (emociones negativas)/cumplir (emociones positivas) con la norma en cuestión. Además, como las emociones están relacionadas con la personalidad del agente, ésta debería considerarse al calcular la relevancia de respetar/violar la norma. Por lo tanto, los agentes que muestran diferentes emociones a los eventos también podrán dar diferentes experiencias a la misma norma.
- *Uso de teorías emocionales*: casi todos los enfoques actuales de los sistemas normativos emocionales se centran en la teoría de la evaluación (*appraisal theory*), principalmente en el modelo OCC. Sin embargo, las teorías dimensionales y las teorías jerárquicas proponen características interesantes que también podrían integrarse en un agente emocional normativo para modelar mejor los problemas emocionales. Por ejemplo, en GENIA3 se utiliza la teoría dimensional [51], mediante el modelo PAD.
- *Descripción de los casos de estudio*: se necesitan más casos de estudio en los cuales la relación entre normas y emociones se represente claramente. Estos casos de estudio deben poder representar diferentes tipos de normas (institucionales, convenciones, de interacción y privadas), diferentes tipos de agentes (con diferentes evaluaciones emocionales,

diferentes personalidades) y la representación de las normas debe hacerse de manera explícita, mediante modelos normativos. Estos casos de estudio también deben ofrecer una representación visual del escenario, con personajes virtuales que permitan una interpretación de las emociones del agente por parte de los usuarios humanos.

Se concluye que quedan aún muchos aspectos por cubrir respecto a la creación de agentes normativo emocionales. En este trabajo se propone un agente normativo emocional (NEA²¹) que integre tanto un modelo emocional como normativo de manera que sea capaz de razonar sobre las normas teniendo en cuenta su estructura emocional. Precisamente el siguiente capítulo abordará los detalles de esta propuesta.

²¹Normative Emotional Agent

NEA: Agente normativo emocional

Este capítulo detalla en profundidad las características del agente normativo emocional (NEA) que se propone en este trabajo. En concreto, se especifica su arquitectura así como las aportaciones para su implementación. El capítulo se organiza como sigue: primero se presenta la representación normativa y el modelo normativo que se toma. A continuación, se muestra la modelización e implementación de las emociones. Finalmente, se muestra el ciclo de un agente NEA, se presentan sus características y sus limitaciones, así como el trabajo que queda aún por hacer en este campo.

3.1 Introducción

En el capítulo anterior se ha visto, por un lado, cómo las normas representan un mecanismo adecuado para garantizar el orden y controlar el comportamiento de los agentes dentro de los sistemas multiagentes. Además, para no minar la autonomía de los agentes, éstos deben ser capaces de razonar sobre las normas y poder decidir si cumplirlas o no. Por otro lado, se ha mostrado la importancia que tienen las emociones en el proceso de toma de decisiones y cómo afectan al resto de agentes, a él mismo así como a su interacción con los demás. Por ello el interés surgido en el desarrollo de un sistema normativo capaz de tener en cuenta tanto las normas como las emociones, ya que beneficiaría y mejoraría la respuesta de los agentes haciéndola más realística. Considerando que éste es un campo aún en investigación, se observa que de las distintas propuestas analizadas, ninguna provee un soporte completo para modelar un sistema normativo emocional que muestre las cuatro relaciones entre normas y emociones que se estableció en el capítulo anterior. Algunas muestran desventajas como permitir modelar las emociones pero no permitir, o no mostrar el proceso de razonamiento normativo. Otras se enfocan solamente en uno de los cuatro tipos de relaciones entre las emociones y normas. Por ello se propone NEA como una extensión de Jason para modelar agentes emocionales normativos capaces de mostrar tres de las cuatro relaciones (la generación de normas privadas lo dejaremos como trabajo futuro a incorporar en el agente).

Existen dos aproximaciones en cuanto al proceso de creación e institución de las normas [78]: *top-down*, donde las normas se diseñan y modelan *off-line*, o bien son creadas por un "agente líder" dinámicamente, que las comunica al resto de agentes; y *bottom-up*, donde la generación de una norma surge de manera que "emerge" del entorno, una vez que es respetada por un grupo numeroso de agentes. Este proceso es útil para la creación de las normas internas que pueden surgir de la observación del entorno. En este trabajo, el proceso de generación de las normas no se contempla y se sigue el modelo *top-down*. En particular, se hace uso de

la creación *off-line*, que quiere decir que las normas presentes se han formalizado e introducido previamente en el entorno. Ahora bien, dicho modelo suele implicar que las normas son impuestas en el agente. Sin embargo, el modelo normativo propuesto e incorporado en NEA pretende que el agente pueda libremente decidir si cumplir o no las normas para simular así la conducta humana, que no acata las normas siempre, sino que razona sobre ellas. Este proceso que denominamos *razonamiento normativo* debe contemplarse durante el razonamiento práctico. Además no sólo se tiene en cuenta los factores normativos, como son la recompensa o sanción de la norma, sino que en ambos procesos el factor emocional se modela y se muestra su aportación en la toma de decisión del agente.

Respecto a la relación entre normas y emociones, existen dos conexiones muy directas entre las normas y las emociones. La primera es cómo las normas y las emociones interfieren en los procesos de toma de decisiones de los agentes. Bagozzi y Pieters [14] estudiaron esta relación y propusieron un modelo de emociones dirigidas por objetivos que toma en cuenta las *emociones anticipadas*, que se refiere a las emociones que se desencadenarán si se logra o no el objetivo asociado. De esta forma, la influencia de las emociones en el proceso de toma de decisiones puede considerarse como un parámetro más de la función de utilidad de los agentes. En un sistema normativo, las emociones intervendrán en la decisión de seguir la norma o no. No sólo por el castigo o la recompensa asociada a la norma, sino también por las emociones positivas o negativas (vergüenza, culpa, orgullo, desilusión, admiración, respeto, consuelo, etc.) que se activarán y que afectarán al agente, a sí mismo y su relación con el resto de los agentes. Por ejemplo, las consecuencias de una emoción negativa se pueden considerar dañinas para el agente, y esta consideración evitará que el agente viole la norma nuevamente. La segunda conexión entre normas y emociones se refiere a cómo las normas sociales influyen en las emociones. Las teorías emocionales permiten establecer un vínculo entre acciones específicas (de uno mismo o de otros) y las emociones que surgen en el individuo. Por lo tanto, si hay un conjunto de emociones que se esperan dado un evento, también se puede determinar una respuesta apropiada a la norma social.

En la propuesta de este trabajo, un agente normativo emocional (NEA) abarca las siguientes características, que se derivan del estudio previo visto en el capítulo anterior respecto a la relación entre normas y emociones:

- *Norm Compliance dilemma*: las emociones se consideran en el proceso de decisión normativa como un parámetro adicional, además de las decisiones racionales (funciones de utilidad).
- *Generación de emociones*: el cumplimiento o violación de las normas genera emociones positivas o negativas en el obediente/infractor de la norma y los observadores de este cumplimiento/violación de la norma.
- *Norm Enforcement*: las emociones se utilizan como mecanismos para hacer cumplir las normas, por lo que las emociones surgen como consecuencias internas positivas (negativas) de un cumplimiento (violación) de una norma social, siendo las emociones el principal resultado de la aplicación de la norma. Por lo tanto, las emociones positivas están relacionadas con las recompensas, mientras que las emociones negativas están relacionadas con las sanciones.
- *Inferencia de norma privada*: las emociones pueden usarse como un mecanismo para promover o inferir normas privadas. A partir de las emociones evaluadas derivadas de ciertas acciones y comportamientos dentro de la sociedad, los agentes pueden construir normas privadas que los ayuden a mejorar su relación dentro del grupo.

Respecto a la implementación se ha decidido usar la plataforma Jason. Algunas de las razones por las que se ha seleccionado son: (i) permite la creación de sistemas multiagentes donde los agentes adoptan la arquitectura BDI, que como se comentó previamente es un modelo ampliamente aceptado para la creación de agentes racionales; (ii) da soporte para una buena comunicación entre los agentes que permite realizar acciones como cooperación y coordinación; (iii) posibilita la modificación del ciclo de razonamiento, permitiendo de manera relativamente sencilla extender el lenguaje de forma que se pueden añadir funciones y modificar las ya provistas, que en nuestro caso permitirán el razonamiento normativo, el procesamiento de las emociones y poder representar la relación que se establece entre normas y emociones; y (iv) Jason es una de las plataformas de agentes más utilizada para la implementación de sistemas multiagente normativos, como se vio en el capítulo anterior.

3.2 Arquitectura de un agente normativo emocional: NEA

3.2.1. Modelo normativo

El agente que se propone parte de la arquitectura BDI que utiliza la plataforma Jason. Sin embargo, ha sido necesario realizar una serie de modificaciones para poder modelar las normas y emociones. En principio se pretendía usar como base normativa el software *v-BDI* que ofrecía Meneguzzi[7]. Sin embargo, finalmente se ha decidido partir desde cero por las complicaciones a la hora de desentrañar el proceso de razonamiento normativo seguido, además de que la representación usada por Meneguzzi difiere en gran medida con la que se pretende usar en este trabajo. En concreto, se pretende que el agente sea capaz de representar, reconocer y razonar sobre las normas y que tenga también en cuenta el aspecto emocional. Por ello primeramente conviene definir las normas y su representación. En este trabajo se usa la lógica deóntica que formaliza las normas como: permisos(*P*), prohibiciones(*F*) y obligaciones(*O*). Así como hacían otras propuestas normativas aquí también se ha decidido simplificar esta visión, asumiendo que aquello no especificado como prohibición está permitido en el entorno, de manera que no se formalizan los permisos.

En el capítulo anterior se presentó un análisis de distintas propuestas de sistemas normativos así como la representación usada para las normas. A partir de ese análisis se ha decidido modelar las normas como percepciones del entorno que los agentes recogen y posteriormente razonan sobre ellas. En concreto, se ha decidido modelarlas como una tupla cuyos atributos se especifican a continuación.

Definición 3.2.1. Modelización de la norma (*n*) como una creencia, tal que si es una obligación $n = obligation(id, A, P, S, R)$; si es una prohibición $n = prohibition(id, P, S, R)$ donde:

- *id*: especifica el nombre de la norma, que es identificativo y único.
- *A*: representa las condiciones de activación de la norma. En nuestro caso se refiere a las creencias que deben ser verdaderas en el agente para que la obligación esté activa.
- *P*: representa la acción sobre la que regula la norma y se refiere específicamente a un *triggering event* que activa el plan que contiene las acciones a obligar/prohibir.
- *S*: expresa la sanción asociada a la norma y se corresponde con un valor numérico dentro del rango [-1,0]. En caso de no haber ninguna se escribe 0.
- *R*: expresa la recompensa asociada a la norma y se corresponde con un valor numérico dentro del rango [0,1]. En caso de no haber ninguna se escribe 0.

Para clarificar se presentan dos ejemplos, uno de cada tipo de norma. Supongamos que hay un agente software que representa a un conductor de un coche. Entre las normas de tráfico establecidas en el entorno hay dos normas, por un lado la obligación $n1$:

$n1 = obligation(use_seatbelt, start_driving, put_seatbelt, ticket, 0)$

indica la presencia de la obligación de usar el cinturón ($use_seatbelt$) cuando se percibe que el agente quiere conducir ($start_driving$). La norma muestra que el evento activador afectado por esta norma es $put_seatbelt$. En caso de incumplir dicha obligación el conductor se vería sancionado con una multa ($ticket$, donde $ticket = -1$), y el 0 indica la ausencia de recompensa. Por otro lado se tiene la prohibición $n2$:

$n2 = prohibition(over_capacity, get_in, ticket, 0)$

representa la prohibición de superar la capacidad del coche ($over_capacity$). Esta norma afecta al evento activador get_in que permite a las personas entrar en el coche. En caso de violar la norma el agente recibe una multa ($ticket = -1$) y el valor cero indica la ausencia de una recompensa.

En nuestra propuesta las normas se encuentran ligadas al entorno, como pasa en la realidad. Así, si te encuentras en Inglaterra las normas que rigen la conducción serán las de dicho país, si te encuentras en España las normas de circulación serán otras. Por tanto las normas y su vigencia vienen establecidas por el entorno y así queda plasmado en nuestra propuesta, donde las normas vigentes asociadas al entorno se modelan como percepciones que recoge el agente. El diseñador se encarga de construir e introducir las normas al entorno. De este modo el agente es consciente de las normas que regulan el entorno en el que se encuentra y además es capaz de diferenciarlas del resto de percepciones como veremos más adelante. Ahora bien, se modelan dos tipos de normas. Por un lado, las prohibiciones, que indican las acciones prohibidas en el entorno, y que no necesitan de condiciones de activación ya que el ser percibidas por el agente indica de por sí que dichas prohibiciones están activas, pues en caso contrario las prohibiciones no serían percibidas por el agente. Por lo tanto, una prohibición se respeta siempre que la acción especificada no se ejecute, y se viola cuando la acción prohibida se ejecuta. Por otro lado, las obligaciones, que expresan el deber del agente a realizar una acción de manera puntual, sí necesitan condiciones de activación para evitar bucles que obliguen a realizar la acción especificada y así poder diferenciar entre la obligación genérica y su instanciación. Por lo tanto, si se trata de una obligación, ésta se cumple siempre que, una vez se active la norma, la acción asociada se ejecuta, y se viola si una vez activada no se ejecuta la acción especificada

En nuestro modelo normativo son los mismos agentes los que monitorizan las normas (el cumplimiento o violación) y responden en consecuencia para penalizar/premiar la acción del agente frente a la norma. Por lo tanto cada vez que un agente cumple/viola una norma, él mismo informa al resto de agentes de su conducta mediante un mensaje. En el caso de recibir una notificación de violación de una norma, el agente receptor envía al agente transgresor un mensaje constatando que siente reproche hacia él. En el caso de ser una notificación de un cumplimiento de una norma el agente receptor enviará al agente que ha respetado la norma un mensaje indicando que siente admiración hacia él. Estas respuestas emocionales son importantes ya que contribuyen en el razonamiento normativo así como en la emoción del agente, como se verá más adelante. Además, para hacer más realista la respuesta de los agentes, se tiene el parámetro *visibilidad* (v) (que se especifica *offline*) que mide el grado de "visibilidad" que el agente receptor tiene respecto a las respuestas del razonamiento normativo que recibe. De manera que dependiendo de este valor, que toma valores dentro del rango $[0,1]$, el agente puede ver todas las respuestas normativas ($v = 1$), no ver ninguna ($v = 0$) o ver algunas sí y

otras no, en base a la probabilidad de visibilidad indicada por ese parámetro (p.e., si $v = 0,8$, tiene un 80 % de probabilidad de ver una respuesta normativa).

Definición 3.2.2. Modelización de la respuesta normativa del agente A a la norma n . Puede decidir violar la norma, en cuyo caso el mensaje que envía notificando al resto de agentes tiene la forma $violation(A, n)$; o decidir cumplirla, en cuyo caso tiene la forma $compliance(A, n)$, donde:

- n : representa la norma en cuestión que ha sido violada o respetada.
- A : especifica el agente que ha violado o respetado la norma n .

Para finalizar, la figura 3.1 muestra un ejemplo de un entorno que sigue el modelo normativo propuesto. Se ve como el diseñador introduce las normas (n_1, n_2 y n_3) *offline* y éstas se modelan como percepciones del entorno. Por su sintaxis es fácil de distinguirlas del resto de percepciones (p_1 y p_2). Se observa que el agente $A1$ ha transgredido la norma n_2 , por lo que informa al resto de agentes ($A2$ y $A3$) de esa violación ($violation(A1, n_2)$). Se aprecia que dado los valores de visibilidad que tengan los agentes observadores ($A2, A3$), estos valores les han permitido observar la transgresión, ya que se observa que en consecuencia, los agentes responden informando a $A1$ que sienten la emoción de reproche ($e_{reproche}$) hacia él.

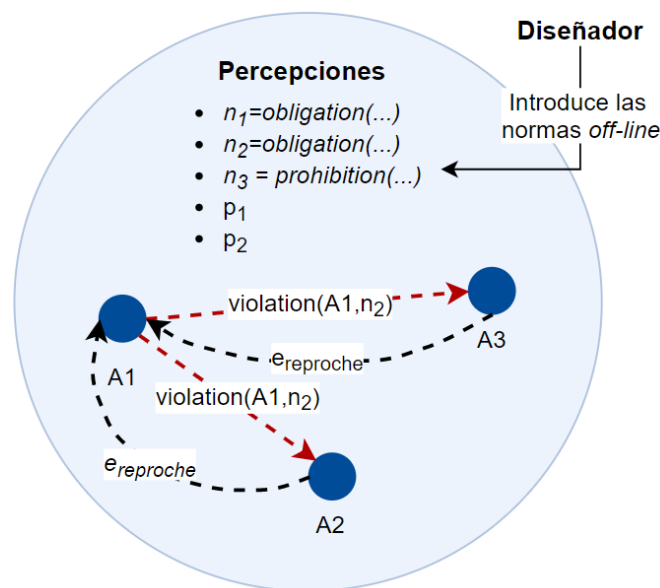


Figura 3.1: Modelado de un sistema normativo emocional

3.2.2. Modelo emocional

Respecto a la parte emocional del agente, se necesita un modelo emocional capaz de modelar y manejar emociones, evaluar el estado del agente para actualizar sus emociones, así como reflejar el efecto que las normas producen en el estado emocional del agente. El modelo emocional que se toma como base en este trabajo es la teoría emocional OCC (*appraisal theory*), que es un modelo psicológico propuesto por Ortony, Clore y Collins [29]. Como se comentó anteriormente en el Capítulo 2, este modelo diferencia 22 emociones como reacción a distintas situaciones (eventos, acciones y objetos). Sin embargo, se ha decidido simplificar el modelo reduciendo el número de emociones que el agente es capaz de manejar, tomando aquellas que

nos conviene para poder expresar la relación entre las normas y emociones. Así, en nuestra propuesta, el agente es capaz de modelar nueve emociones categorizadas en: (i) *emociones positivas* (felicidad, orgullo, gratificación y admiración); (ii) *emociones negativas* (tristeza, remordimiento, reproche y enojo); (iii) y una *emoción neutra*, calma, desde la cual parte el agente como emoción inicial. Ahora bien, estas emociones las clasificamos también en otras dos categorías según a quién van dirigidas: "*Agente observador*", en donde se recogen las emociones que el agente genera hacia otros agentes como respuesta a las acciones que observa de los demás, y "*Agente mismo*", que agrupa las emociones que el agente genera sobre sí mismo como respuesta a sus propias acciones. La tabla 3.1 muestra un resumen de las emociones soportadas por el modelo emocional adoptado por el agente NEA. Además, cabe destacar que entre las emociones que soporta el agente diferenciamos también un subgrupo de emociones a las que denominamos: *emociones anticipadas*. Se trata de emociones cuya generación está relacionada con las normas, es decir, son de carácter normativo. Se dividen en dos grupos: *emociones normativas externas (ENE)*, que son las emociones recibidas por parte del resto de agentes como respuesta a las acciones del agente (admiración, reproche), y las *emociones normativas internas (ENI)*, que son las emociones que el agente ha despertado en sí mismo, como consecuencia de cumplir o infringir la norma (orgullo, remordimiento). Estas emociones contribuyen en el proceso de razonamiento normativo y en la actualización del estado emocional del agente. Señalar que para simplificar el modelo emocional se ha decidido asumir que en caso de una violación de una norma la respuesta de los agentes será siempre la emoción de reproche y la respuesta del mismo agente hacia él sería remordimiento. En caso de cumplimiento de la norma, los agentes sólo pueden expresar admiración sobre el agente, y el agente sobre sí mismo expresará orgullo.

	Emociones		E. Anticipadas
	Ag. Observador	Ag. Mismo	
Positivas	gratificación	felicidad	
	admiración	orgullo	
Negativas	reproche	remordimiento	
	enojo	tristeza	
Neutra	calma		

Tabla 3.1: Emociones soportadas por un agente NEA

Para implementar este modelo emocional se ha decidido emplear valores numéricos y tener una *base de emociones (BE)* que guarde un registro de las emociones soportadas por el agente y los parámetros necesarios para el cálculo de las emociones. Para ello se hace uso de la Base de Creencias que nos proporciona Jason para guardar como creencias: los parámetros necesarios para el cálculo de las emociones (que el diseñador se encarga de introducir *offline* inicializándolos como corresponde), el estado emocional y las emociones externas recibidas.

El *estado emocional* (θ_{ES}) del agente se representa como un valor que oscila dentro del rango $[-1,1]$, donde cada valor dentro de ese rango se corresponde con una emoción. En la figura 3.2 se muestra dicha asociación: $tristeza \in [-1, -0,75]$, $remordimiento \in [-0,75, -0,25]$, $calma \in [-0,25, 0,25]$, $orgullo \in (0,25, 0,75]$ y $felicidad \in (0,75, 1]$. Se ha decidido dar un rango menor que al resto de emociones a las emociones de tristeza y felicidad, ya que las hemos considerado como extremas. Así, dado el estado emocional de un agente, este evoluciona bien positivamente (hacia felicidad) o negativamente (hacia tristeza.). El proceso del cálculo del estado emocional se explica con detalle más adelante en la siguiente sección. Los parámetros necesarios para el cálculo de las emociones son: las emociones anticipadas (e_{ad} , e_{org} , e_{rep} y e_{rem}), ri (*reputation*

importance) y por último el estado emocional (θ_{ES}) actual del agente. Los detalles concretos de estos valores se explicarán también en la siguiente sección.

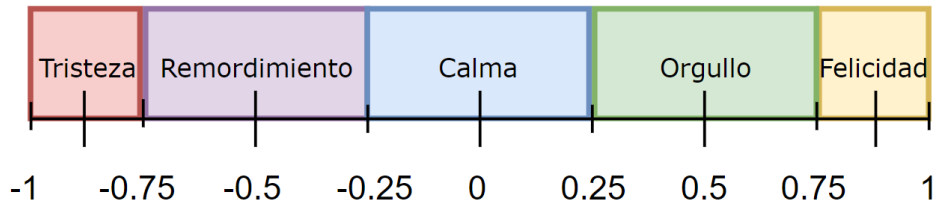


Figura 3.2: Modelado de las emociones soportadas por un agente NEA

3.2.3. Ciclo de un agente NEA

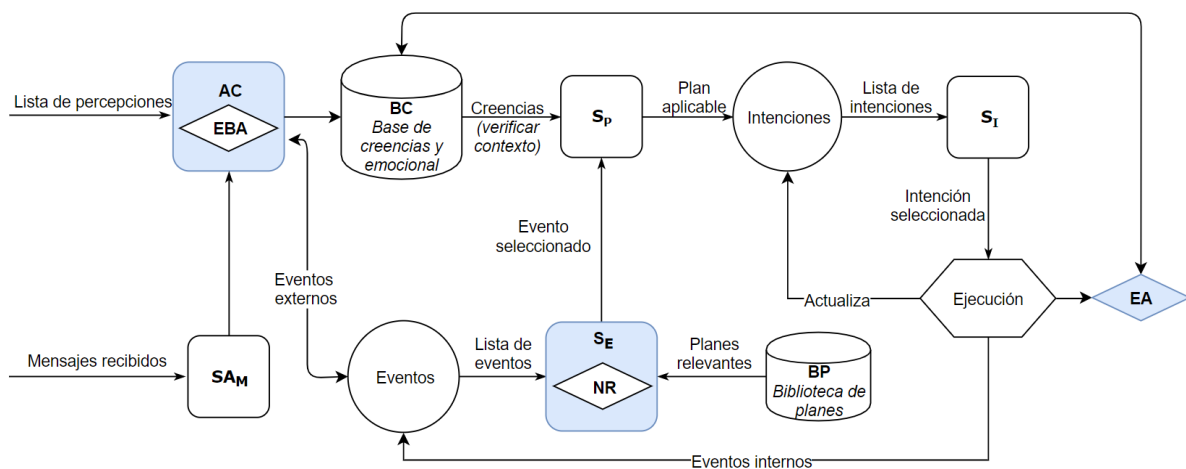


Figura 3.3: Ciclo de razonamiento propuesto para un agente normativo emocional (NEA) en Jason

El ciclo de razonamiento del agente NEA que hemos desarrollado en este trabajo consiste, en forma breve, en lo siguiente. En primer lugar el agente recoge las percepciones del entorno, entre las cuales se encuentran las normas vigentes del entorno que le afectan (ver figura 3.3). Las percepciones pasan por la función de *actualización de creencias* (AC), que ha sido modificada de tal manera que además de actualizar la base de creencias almacenando las normas, también se actualiza la base de emociones. La función *emotional base actualization* (EBA) se encarga de actualizar los parámetros emocionales y responder a las transgresiones y cumplimientos de normas recibidos (en las respuestas normativas) con la correspondiente emoción anticipada que se genere en él (reproche, admiración).

Después de la actualización de eventos (debido a la adición/eliminación de percepciones) y la base de creencias, se procede a la selección de un evento. Esto lo realiza la función de *selección de evento* (S_E) que ha sido modificada de manera que la lista de eventos se ordena de acuerdo a la prioridad asociada a cada evento y se escoge aquel con mayor prioridad. En caso de que el evento seleccionado se corresponda con una norma se procede a realizar el razonamiento normativo. A diferencia de la mayoría de propuestas que esperaban a la selección del plan para realizar el razonamiento normativo, nuestra propuesta propone realizarlo durante la selección del evento. La función de *selección de evento* (S_E) devuelve un evento tal que, si su acción no está afectada por una norma, se continúa con el ciclo. Sin embargo, si su acción está afectada por

una norma, se procede a realizar el razonamiento normativo, que la función *normative reasoning* (NR) se encarga de ejecutar y se explicará en detalle más adelante.

A continuación se pasa a la *selección del plan* (S_p) que, de entre los planes aplicables, selecciona uno. El plan seleccionado actualiza el conjunto de intenciones del agente y la función *selección de intenciones* (S_I) escoge la intención más apropiada y se procede a su ejecución. Posteriormente se procede a actualizar las intenciones, se elimina el plan ejecutado del conjunto de intenciones y la función *emotion actualización* (EA), también propuesta en este trabajo, se encarga de actualizar el estado emocional del agente.

En las secciones que vienen a continuación se explicará con más detalle las funciones del ciclo de razonamiento NEA que hemos propuesto como extensión al ciclo de razonamiento de Jason para que los agentes puedan dar soporte a las emociones y normas.

3.2.4. Actualización de la base de emociones (EBA)

La función *emotion base actualización* (EBA) se encarga de actualizar la base de emociones y dar una respuesta emocional a las acciones normativas que el agente ha observado (i.e., le han sido visibles, de acuerdo a su parámetro de visibilidad (ver sección 3.2.1). La base emocional almacena: parámetros estáticos (e_{org} , e_{rem} y ri) que el diseñador establece *offline* (donde e_{org} representa el grado de orgullo del agente; e_{rem} su grado de remordimiento; y ri la importancia de su reputación), y parámetros dinámicos (v , e_{ad} , e_{rep} y θ_{ES}) que el diseñador inicializa y el agente va actualizando durante su ciclo de vida (donde v representa la visibilidad del agente respecto a las respuestas normativas que recibe el entorno; e_{ad} representa las emociones de admiración recibidas; e_{rep} recoge las emociones de reproche recibidos y θ_{ES} es el estado emocional actual del agente). Dichos parámetros se modelan en el agente como creencias dentro de la base de creencias pero que el agente es capaz de distinguir fácilmente gracias a la sintaxis empleada.

De la lista de percepciones que llegan la función de actualización de creencias, la función EBA toma aquellas relacionadas con las emociones, es decir recoge y almacena como creencias las emociones de admiración (e_{ad}) y reproche (e_{rep}) que ha generado en el resto de agentes en caso de haber violado/respetado una norma. Estas emociones le llegan a través de los mensajes y de igual manera el agente recibe también, las acciones de transgresión/cumplimiento de las normas del resto de agentes. Como ya comentamos anteriormente, cuando una agente recibe las respuestas normativas de otros agentes, procede según su parámetro de visibilidad, y en caso de que la respuesta normativa le sea visible, procede a enviar un mensaje informando al agente en cuestión (incluido en la respuesta normativa) de la *emoción normativa externa* (e_{ENE}) que dicha respuesta normativa le ha generado.

3.2.5. Actualización del estado emocional del agente (EA)

Como se comentó anteriormente, la emoción que el agente experimenta se denomina *estado emocional* (θ_{ES}) y toma valores dentro del rango $[-1, 1]$. El estado emocional se ve afectado por las consecuencias emocionales que implica violar/respetar una norma (emociones anticipadas). Nuestro agente NEA parte del estado emocional calma, cuyo valor es 0. La función *emotion actualización* (EA) descrita en la ecuación 3.1 se encarga de actualizar dicho estado emocional para cada ciclo de vida del agente (ver figura 3.3). La función en cuestión considera los atributos: emociones normativas externas (admiración, reproche), emociones normativas internas (orgullo, remordimiento), y por último el estado emocional actual del agente. Los datos

necesarios para el cálculo de la función se obtienen de la base de emociones, de manera que la actualización del estado emocional del agente NEA se calcula como:

$$\theta_{ES}^{t+1} = \min(1, \max(\theta_{ES}^t + \frac{e_{ad} - e_{rep}}{t_{NAg} - 1} \cdot ri + e_{ENI}, -1)) \quad (3.1)$$

donde:

- θ_{ES}^t : parámetro numérico dentro del rango [-1,1] que se corresponde con el estado emocional del agente en el tiempo t .
- θ_{ES}^{t+1} : parámetro numérico dentro del rango [-1,1] que se corresponde con el estado emocional del agente en el tiempo $t + 1$.
- e_{ad} : representa el número de emociones de admiración recibidas por parte de agentes externos en el ciclo actual (instante t), siendo $0 \leq e_{ad} \leq (t_{NAg} - 1)$.
- e_{rep} : representa el número de emociones de reproche recibidas por parte de agentes externos en el ciclo actual (instante t), siendo $0 \leq e_{rep} \leq (t_{NAg} - 1)$.
- t_{NAg} : representa el número total de agentes del sistema. Además se cumple que para el instante t , $e_{ad} + e_{rep} \leq (t_{NAg} - 1)$
- ri (*reputation importance*): parámetro numérico dentro del rango [0,1] que mide la importancia que da el agente a las respuestas emocionales del resto de agentes, viene establecido previamente por el diseñador.
- e_{ENI} : parámetro que toma el valor: (i) 0 si la acción ejecutada por el agente no implica ninguna norma; (ii) e_{org} , si la acción implica el cumplimiento de la norma; (iii) e_{rem} , si la acción realizada implica violar la norma. Por tanto este parámetro representa la importancia que el agente le da al orgullo/remordimiento en caso de cumplir/infringir la norma. El valor de las variables $e_{org} \in [0, 1]$ y $e_{rem} \in [0, 1]$ queda previamente establecido por el diseñador.

Para clarificar se presentan dos ejemplos:

Ejemplo 1

$$\theta_{ES}^t = -0.9 \quad t_{NAg} = 3$$

En un entorno con tres agentes ($A1, A2, A3$), en el instante t el agente $A1$ se encuentra muy triste ($\theta_{ES}^t = -0.9$) y realiza una acción que implica el cumplimiento de una norma. La base emocional del agente $A1$ refleja que al agente le importa bastante seguir las normas y también valora mucho su reputación ($e_{org} = 1, e_{rem} = 1$ y $ri = 1$). Como hemos comentado, el agente enviará al resto de agentes un mensaje indicando que ha cumplido la norma ($compliance(A1, n_1)$). Los agentes $A2$ y $A3$, en función de su respectivo parámetro de visibilidad (definido por el diseñador) tendrán una probabilidad determinada de observar o no el cumplimiento de esa norma, y en el caso de observarlo, le enviarán su mensaje de admiración. Por tanto, podemos diferenciar aquí tres casos, dependiendo del número de agentes que lo hayan observado y, por ende, le han comunicado su admiración al respecto.

- a) Nadie observa su respeto a la norma, por lo que: $e_{ad} = 0, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(-0.9 + \frac{0-0}{2} \cdot 1 + 1, -1)) = -0.4 \in \text{Remordimiento}$$

b) La mitad de los agentes del sistema le han observado: $e_{ad} = 1, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(-0.9 + \frac{1-0}{2} \cdot 1 + 1, -1)) = -0.15 \in \text{Calma}$$

c) Todos los agentes del sistema le han observado: $e_{ad} = 2, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(-0.9 + \frac{2-0}{2} \cdot 1 + 1, -1)) = 0.1 \in \text{Calma}$$

Ejemplo 2

$$\theta_{ES}^t = 0.55 \quad t_{NAg} = 3$$

En un entorno con tres agentes ($A1, A2, A3$), en el instante t el agente $A1$ se encuentra bastante orgulloso de sus actos ($\theta_{ES}^t = 0.55$) y realiza una acción que implica la violación de una norma. La base emocional del agente $A1$ refleja que el agente no le da mucha importancia a las normas y da algo de importancia a "lo que dirán los demás" (su reputación) ($e_{org} = 0.5, e_{rem} = 0.5$ y $ri = 0.5$). Al igual que antes, diferenciaremos tres casos:

a) Nadie observa su violación a la norma, por lo que: $e_{ad} = 0, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(0.55 + \frac{0-0}{2} \cdot 0.5 - 0.5, -1)) = 0.3 \in \text{Orgullo}$$

b) La mitad de los agentes del sistema le han observado: $e_{ad} = 1, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(0.55 + \frac{0-1}{2} \cdot 0.5 - 0.5, -1)) = 0.175 \in \text{Calma}$$

c) Todos los agentes del sistema le han observado: $e_{ad} = 2, e_{rep} = 0$

$$\theta_{ES}^{t+1} = \min(1, \max(0.55 + \frac{0-2}{2} \cdot 0.5 - 0.5, -1)) = 0.05 \in \text{Calma}$$

En el primer ejemplo se observa como el estado emocional del agente evoluciona positivamente debido a que para él es muy importante el cumplir con las normas y el estado de su reputación, y por lo tanto le afecta emocionalmente. Por ello se observa que aunque nadie vea que ha cumplido con la norma igualmente él ya se encuentra un poco mejor ya que se aleja de la emoción tristeza y siente remordimiento. Cuando la mitad de los agentes le observa, queda plasmado en su emoción el efecto positivo que ha tenido en él, pasando de tristeza a sentirse algo calmado. Y finalmente cuando recibe la respuesta de todos los agentes pasa a sentir una calma firme ya que el valor obtenido se acerca más a cero, que en el caso anterior.

En el segundo ejemplo se observa que a pesar de que el estado emocional del agente evoluciona negativamente, aún cuando todos los agentes perciben su violación y recibe el reproche de todos, su estado emocional aún se encuentra "positivo" ya que la calma que experimenta no se encuentra en el rango negativo. Por lo tanto se observa que los parámetros que hemos establecido en él, hacen que la evolución negativa sea más progresiva.

3.2.6. Razonamiento normativo

La función *normative reasoning* (NR) es una función de utilidad que realiza el razonamiento práctico sobre la norma, devolviendo un valor denominado θ_{will} que muestra la decisión que ha tomado el agente respecto a la norma activa en cuestión, (i.e. cumplir o infringir la norma). La construcción de la función se basa en el trabajo realizado por Criado et al. [2], que a su vez toma como base el trabajo de Elster [21]. La propuesta de agentes $n - BDI$ de Criado et al. no consta de una representación explícita de las emociones ni de su razonamiento. Además, tal y como los autores comentan, su agente $n - BDI$ no es emocional sino que se centra más en la parte normativa y el razonamiento normativo del agente. Por su parte, Elster en su trabajo determina qué factores influyen en la voluntad del agente a cumplir con las normas, que pueden resumirse en tres: (i) interés personal; (ii) las expectativas respecto a la decisión (sanciones y recompensas); y (iii) factores emocionales, que nosotros entendemos como emociones anticipadas. Sin embargo, esta propuesta limita la contribución de las emociones en una sola variable, cuando por el análisis presentado anteriormente se observa que las emociones afectan mucho más, por esto nuestra propuesta modifica estos criterios para poder plasmar la interacción entre normas y emociones y mostrar cómo se afectan mutuamente.

La función de utilidad que proponemos para realizar el proceso de razonamiento normativo tiene en cuenta los siguientes atributos: las consecuencias normativas (sanción y recompensa), las emociones anticipadas (e_{ENE} y e_{ENI}) y la importancia que el agente da a la opinión del resto de agentes sobre sus acciones (*reputation importance*, ri). Las sanciones y recompensas vienen dadas por la norma, las emociones normativas internas (e_{org} y e_{rem}) y ri son parámetros estáticos que el diseñador inicializa *offline*. Sin embargo las emociones normativas externas (e_{ad} y e_{rep}) nos son desconocidas ya que se obtienen como consecuencia de la decisión del agente a respetar o violar la norma. Por esto a partir de la idea de Elster que calcula las expectativas sobre las sanciones y recompensas, se ha decidido tomar esta idea y aplicarla sobre las emociones normativas externas de manera el agente pueda calcular la probabilidad de que su acción de violar/respetar una norma sea captada por el resto de agentes del entorno. Para ello primero cabe puntualizar que se realiza una asunción por la que siempre que un agente viole una norma, la respuesta del resto de agentes será siempre reproche, siempre que sea para ellos visible esta transgresión. Del mismo modo en caso de respetar una norma la respuesta de los agentes será admiración en el caso de que haya sido visible para ellos. Es decir, no se contempla una respuesta negativa como lo es el reproche frente a la acción de respetar la norma, ni una respuesta positiva como lo es la admiración frente a la acción de transgredir una norma.

La probabilidad de las emociones normativas externas (P_E), i.e. la probabilidad de que la acción de violar/cumplir una norma sea vista por el resto de agentes del sistema, se calcula como se muestra en la ecuación 3.2. El diseñador define una ventana Δ que representa el número de registros que se guardarán de las emociones normativas externas. Así P_E se calcula como la media de la suma de los porcentajes que se recogen en la ventana Δ . Estos porcentajes se calculan como el número de emociones normativas recibidas (e_{ENE}) en el instante t , partido entre el número de agentes del sistema menos el agente en cuestión (t_{NAg-1}).

$$P_E = \frac{\sum ((e_{ENE})_{t_i} / (t_{NAg} - 1))}{\Delta}, \quad \text{siendo } i = -1, -2, \dots, -\Delta \quad (3.2)$$

Por ejemplo, se tiene un sistema con tres agentes ($t_{NAg} = 3$), donde se ha definido una ventana $\Delta = 3$. Esto quiere decir que cada agente almacena en su base de creencias un registro de las tres últimas respuestas de las emociones normativas externas recibidas. Por ejemplo, el registro que guarda el agente A1 que se muestra en la tabla 3.2 y con esos datos obtenemos que

la probabilidad que observen y respondan ante la decisión del agente A1 sobre una norma, es del 83 %.

t-1	t-2	t-3
$\frac{1}{2}$	$\frac{2}{2}$	$\frac{2}{2}$
50 %	100 %	100 %

Tabla 3.2: Registro de las e_{ENE} recibidas en el instante t

$$P_E = \frac{0.5 + 1 + 1}{3} = 0.83 \rightarrow 83 \%$$

Respecto a la función de utilidad que se emplea para el razonamiento normativo, nuestra propuesta es usar dos parámetros intermedios, $\theta_{fulfill}$ y $\theta_{violation}$ que calculan el beneficio que aportaría respetar y violar la norma respectivamente. Las ecuaciones 3.3 y 3.4 especifican su cálculo. De forma que el valor que toma θ_{will} se obtiene en función de estos parámetros tal y como se indica en la ecuación 3.5. Así, si el valor de θ_{will} es uno, el agente decide cumplir con la norma, si es -1 se decide violar la norma, y por último, si el valor es cero, de manera aleatoria se decide si violar o no ya que el cero indica que ambas opciones son igual de beneficiosas.

$$\theta_{fulfill} = |R + e_{org}| \cdot P_E \cdot ri \quad (3.3)$$

$$\theta_{violate} = |S + e_{rem}| \cdot P_E \cdot ri \quad (3.4)$$

$$\theta_{will} = \begin{cases} 1, & \text{si } \theta_{fulfill} > \theta_{violate} \\ -1 & \text{si } \theta_{violate} > \theta_{fulfill} \\ 0 & \text{si } \theta_{violate} = \theta_{fulfill} \end{cases} \quad (3.5)$$

Para clarificar se presenta un ejemplo, en concreto partimos del agente del ejemplo 1, caso a, que se comentó en la sección anterior. El agente en cuestión se encuentra razonando sobre la norma $n2$ (que se presentó en la sección 3.2.1 *Modelo normativo*):

$$n2 = prohibition(over_capacity, get_in, ticket, 0) \text{ (donde } ticket = -1)$$

El agente valora positivamente la recompensa emocional de respetar una norma ($e_{org} = 1$) y teme a la sanción emocional que surge de violar la norma ($e_{rem} = 1$), por lo que tenderá a respetar las normas. Además le da bastante importancia a su reputación ($ri = 1$) por lo que le afectan las emociones anticipadas que despierta en los otros agentes. El valor del parámetro $P = 0.83$ (que se toma del ejemplo anterior), por lo que, después de realizar los cálculos necesarios se tiene que el valor de $\theta_{will} = 1$ por lo que el agente ha decidido respetar la norma, tal y como se esperaba dados los parámetros que tiene asignados.

$$\theta_{fulfill} = |0 + 1| \cdot 0.83 \cdot 1 = 0.83$$

$$\theta_{violate} = |-1 + 1| \cdot 0.83 \cdot 1 = 0$$

$$\theta_{will} = 1 \text{ ya que se cumple que } 0.83 > 0$$

3.3 Discusión

Este capítulo se ha centrado en presentar la arquitectura de un agente emocional normativo (NEA) así como en detallar los componentes propuestos para extender el ciclo de razonamiento Jason, como son el modelo emocional y normativo para que el agente sea capaz de dar soporte a las normas y emociones. Una de las contribuciones del agente NEA es que modela tres de las cuatro relaciones que se estableció entre las normas y emociones (ver Figura 3.4). La primera relación expresaba la aportación de las emociones a la hora de decidir si cumplir o no la norma. Esto queda plasmado en la función *normative reasoning* (NR) donde uno de los parámetros son las emociones anticipadas. Respecto a la segunda relación, que mostraba que las consecuencias normativas de cumplir o infringir la norma quedaban reflejadas en la emoción del agente, en nuestra propuesta esta relación queda modelada en el agente en la función *emotion actualization* (EA), donde se tienen en cuenta las emociones anticipadas externas e internas que afectan al estado emocional del agente. En cuanto a la tercera relación, que expresa que las emociones ayudan a mantener las normas socialmente, queda reflejado en la representación de las emociones anticipadas que el agente NEA es capaz de modelar y responder así a las transgresiones de otros agentes.

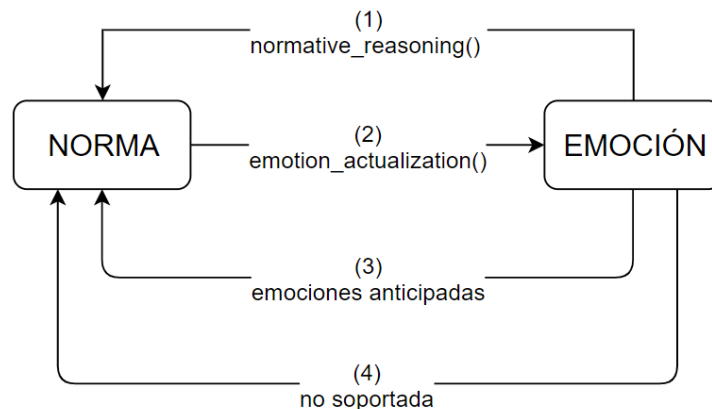


Figura 3.4: Soporte de un agente NEA a las cuatro relaciones entre normas y emociones

Respecto a las limitaciones que presenta, el agente no da soporte a la cuarta relación por ser una cuestión que requiere un estudio intensivo acerca de la moral y otros aspectos del agente, y que este trabajo no abarca. Así mismo el proceso de generación y aceptación de normas nuevas que emergen del entorno no se soporta. Respecto al aspecto emocional del agente, éste se ha simplificado bastante y no se ha modelado el estado de ánimo, ni tampoco la personalidad. No obstante, el agente NEA es capaz de representar agentes con distintos roles, gracias a los parámetros e_{org} , e_{rep} y ri , de modo que, como se ha visto en los ejemplos, podemos representar a agentes muy preocupados por sus acciones normativas, así como a agentes a los que no les importe la opinión de los demás, o no les afecte emocionalmente el resultado de sus acciones normativas.

En el siguiente capítulo se muestra la implementación del agente normativo emocional y se presenta un caso de estudio así como las conclusiones obtenidas.

CAPÍTULO 4

Caso de estudio

En este capítulo se presenta un caso de estudio sencillo, compuesto por agentes normativos emocionales cuya estructura se ha mostrado en el capítulo anterior. El capítulo se organiza como sigue: primero una breve introducción comentando los distintos casos de estudio del estado del arte, respecto a agentes normativos emocionales. A continuación, se detalla el caso de estudio propuesto en este trabajo, así como el proceso de implementación y el código de las funciones necesarias que se han llegado a implementar. Finalmente se expone el trabajo futuro que queda por realizar.

4.1 Caso de estudio

4.1.1. Introducción

En el análisis del estado del arte realizado en el capítulo 2, hemos observado que algunos trabajos han basado sus experimentos en casos de estudio sobre el control de la agresión, como el propuesto por Conte y Castelfranchy [69]. En su ejemplo, los agentes realizan un conjunto de acciones básicas para sobrevivir en una situación de escasez de alimentos (por ejemplo, moverse, comer o atacar a un agente mientras come). Cada agente tiene asignada su comida y todos los agentes siguen una estrategia normativa para controlar la agresión: no atacan a los agentes que comen su propia comida (esta es la norma institucional, denominada norma "*finder-keeper*"). Además, los agentes siguen una estrategia utilitaria para controlar la agresión: no atacan a agentes cuya fuerza es mayor que la suya. El resultado de este caso de estudio mostró que la estrategia normativa redujo la agresión (es decir, el número de ataques) mucho más que la estrategia utilitaria [69].

Staller et al.[56] propone un caso de estudio basado en [69] donde también se considera la evaluación de la preocupación, de manera que el estado óptimo de alimentación se considera una preocupación básica para los agentes. Por lo que siempre que no se satisfaga esta preocupación, los alimentos se consideran relevantes. Por lo tanto, los "agentes emocionales normativos" deciden si obedecer o no la norma institucional (i.e, la norma "*finder-keeper*"), basándose en la solidez de sus preocupaciones sobre el estado óptimo de alimentación y su interés por cumplir con las normas. En [64] se propuso un experimento similar, pero empleando las emociones para representar diferentes tipos de agentes: alegres, resentidos, doloridos o enojados. Todos estos experimentos muestran que los agentes emocionales normativos en un sistema normativo social son más eficientes que los agentes normativos a secas, porque

los primeros terminaron con mayor fuerza y una tasa de ataque menor, teniendo así un mejor rendimiento que los agentes solo normativos.

En [66], los autores validan su marco OP-RND-E con el caso de estudio “Preparación de un examen y su moderación” (EPMP), donde compararon los agentes normativos racionales con los agentes normativos emocionales. Este caso de estudio intenta determinar las acciones y las emociones de un profesor al ejecutar el proceso de preparación y envío del documento de examen al comité de exámenes.

Finalmente, la propuesta de [57] fue probada en un escenario inspirado en la ley antitabaco existente en bares y restaurantes en muchos países europeos. En este escenario, el avatar del usuario está sentado con otros agentes dentro de un bar donde está activa la norma “No fumar en el recinto”. Después de una conversación inicial, que indica qué agentes son amigos y cuáles son completamente desconocidos, uno de los agentes comienza a fumar (porque considera que su objetivo de fumar es más importante que el de respetar la norma). Los agentes restantes reaccionan emocionalmente ante esa violación de la norma. Los autores evaluaron diferentes versiones del caso de estudio, cambiando la importancia de las normas y modificando la configuración de los miembros del grupo de agentes (número de amigos y extraños). Su modelo fue capaz de generar emociones en caracteres sintéticos similares a los experimentados por los humanos en situaciones similares.

4.1.2. Caso de estudio general: estación de taxis

En este trabajo proponemos un caso de estudio, relativamente sencillo, que permite observar las respuestas del agente NEA descrito en el capítulo anterior. El escenario propuesto para este trabajo simula una situación social en una estación de taxi, donde los taxis recogen a los posibles clientes. Los clientes representan a un grupo de personas que desean viajar juntas en el mismo taxi. La estación está compuesta con un número T de taxistas, donde cada uno tiene asignado un vehículo con una capacidad. La estación esta compuesta por una fila por la que tiene que pasar los taxistas para poder recoger a un cliente, de manera que sólo el taxista situado a la cabeza de la cola puede recoger a un cliente. Ahora bien, la carretera es lo suficientemente ancha por lo que permite que un taxista pueda colarse si así lo decide. El entorno se encuentra regulado por las siguientes normas:

- El taxista debe colocarse en la fila y respetar el turno de llegada de los taxis para poder recoger clientes. Por tanto, tiene prohibido saltarse su orden en la fila.

n1: prohibition(respectLine, skipLine, -1, 0)

- El taxista tiene prohibido superar la capacidad máxima del vehículo que conduce. En caso de no poder atender al cliente asignado por falta de capacidad, deberá dejar el primer puesto de la fila y hacerla de nuevo. Este acto de rehacer la cola queda registrado por la estación de taxis, de manera que al final los taxistas tienen una recompensa monetaria proporcional al número de veces que les tocó rehacer la cola. Por ello la recompensa de esta norma toma el valor 0.7.

n2: prohibition(respectCapacity, exceedCapacity, -1, 0.7)

- El taxista está obligado a descansar al menos 30 minutos después de una jornada de trabajo de 8 horas.

n3: obligation(respectRest, overWork, rest, -1, 0)

Se modelan tres roles de taxistas, en función del valor de los parámetros emocionales: e_{org} , e_{rem} y ri . Así podemos modelar un *taxista respetuoso* (t_{res}), que da bastante importancia a las consecuencias normativas; un *taxista transgresor* (t_{tra}) que no da importancia a las consecuencias normativas; y por último un *taxista dubitativo* (t_{dub}) que es moderado respecto a la importancia que da a las consecuencias normativas.

	T. Respetuoso	T. Dubitativo	T. Transgresor
e_{org}	1	0.5	-1
e_{rem}	1	0.5	-1
ri	1	0.5	-1

Tabla 4.1: Tipos de Roles de taxistas

El objetivo fundamental de los taxistas es obtener la mayor ganancia posible. Los taxistas pueden encontrarse en dos tipos de estados: trabajando ($state(working)$), que indica que el agente se encuentra actualmente realizando un trayecto con un cliente, i.e. está trabajando; y conduciendo ($state(driving)$) que indica que el agente no se encuentra en la fila, ni con un cliente, pero que está en condiciones de trabajar (por ejemplo cuando vuelve de un trayecto de dejar al cliente).

4.1.3. Caso particular: estación con tres taxis

A continuación se presenta la implementación de una simplificación del caso de estudio expuesto en la sección anterior. El entorno de la estación de taxi que implementamos se encuentra regulado por la siguiente norma:

- El taxista tiene que colocarse en la fila y respetar el turno de llegada de los taxis para poder recoger clientes. Por tanto, tiene prohibido saltarse su orden en la fila.

$n1: prohibition(respectLine, skipLine, -1, 0)$

La estación está compuesta por tres agentes taxistas ($T1, T2, T3$), cuyos vehículos tienen una capacidad de 8 (es decir, que como máximo pueden llevar a clientes que sean un grupo de 8 personas o menos). La figura 4.1 muestra una instantánea del estado inicial de la estación de taxis, donde la primera columna representa el estado trabajando ($working$), que indica que los agentes que se encuentren en dicha columna se encuentran actualmente trabajando ($state(working)$). La segunda columna representa el estado conduciendo ($driving$), que indica que los agentes presentes en dicha columna se encuentran en ese instante en el estado conduciendo ($state(driving)$). La tercera columna se corresponde con la línea que deben guardar los taxistas. Se muestra que los agentes $T1, T2$ y $T3$ se encuentran en la cola ocupando las posiciones 1, 2 y 3 respectivamente. Por último la cuarta columna muestra el cliente que necesita coger un taxi (en caso de no quedar más clientes la celda queda vacía). El cliente se modela como un vector de dos números donde el primer número representa el número de personas que aglutina ese cliente (i.e personas que quieren viajar juntas en el mismo taxi, a ser posible), y el segundo indica el precio que han de pagar por el trayecto. Por ejemplo, el cliente que se muestra en la figura 4.1, indica que son 3 personas y que el costo del trayecto que desean hacer es de 11.





Working (Trabajando)	Conduciendo (Driving)	Línea de taxi (Taxi line)	Cientes (Clients)
		 T1	 Client (3,11)
		 T2	
		 T3	

Figura 4.1: Instantánea del entorno del caso de estudio

4.2 Implementación

Para la implementación de este caso de estudio se ha empleado Jason; y para la creación del entorno, se ha empleado Java. Ahora bien, se ha extendido Jason aplicando ciertos cambios necesarios para dar soporte a la arquitectura NEA comentada en el capítulo anterior. De este modo se han tenido que modificar algunas funciones de Jason y crear nuevas. A continuación se presenta los agentes taxistas y las modificaciones que se han realizado sobre Jason.

4.2.1. Código del taxista

El taxista parte con una base de creencias inicial de dos creencias: $line(x)$, que contiene la posición inicial x del taxista en la fila (por ejemplo, $line(2)$ indica el segundo lugar en la fila); y $state(s)$, que representa el estado actual s (por ejemplo $state(working)$ indica que el agente se encuentra trabajando). Además de estas creencias iniciales, se tienen las creencias de la base de emociones que, como se comentó en el capítulo anterior, se almacenan también en la base de creencias. De este modo los parámetros emocionales necesarios para modelar las emociones en el agente también forman parte de las creencias iniciales del agente. En concreto:

- $emotions[...]$: lista que guarda las emociones que soporta y modela el agente NEA (necesario para luego asignar la emoción al valor obtenido por θ_{ES}).
- $emotionalState(0)$: creencia que almacena el estado emocional θ_{ES} del agente. Inicialmente su valor es cero porque el agente parte de la emoción calma que se corresponde con dicho número.
- $pE[0, 0, 0]$: lista que guarda las últimas Δ respuestas de las emociones normativas externas, siendo $\Delta = 3$ en este ejemplo. Estos datos son necesarios para poder calcular la probabilidad de que la acción del agente sea vista.
- ri, v : estas creencias almacenan, respectivamente, los valores de "reputation importance" (ri) y la visibilidad del agente a las acciones del resto de agentes (v).

- *admiration*, *reproach*, *pride*, *remorse*: creencias que guardan respectivamente los valores dinámicos de e_{ad} (cantidad de emociones de admiración recibidas en el instante t) y e_{rep} (cantidad de emociones de reproche recibidas en el instante t); y los valores estáticos de e_{org} (parámetro que mide cuánto orgulloso se siente el agente de él mismo por respetar la norma) y e_{rem} (parámetro que mide cuánto remordimiento siente el agente sobre él mismo por transgredir la norma) .

A modo de ejemplo, la figura 4.2 muestra las creencias iniciales de un agente NEA taxista.

```

5 // Initial beliefs
6 line(1).
7 state(working).
8
9
10 //Emotional beliefs
11 emotions["tristeza", "remordimiento", "Calma", "Orgullo", "Felicidad"].
12 emotionalState(0). // Estado emocional
13 pE[0,0,0].
14 ri(1). // Reputation importance
15 v(0.7).
16
17 //Anticipated emotions
18 admiration(0). //ENE - external normative emotions
19 reproach(0).
20
21 pride(1). //INE - internal normative emotions
22 remorse(1).
23
24 //Norms safe as beliefs
25 prohibition(respectLine, skipLine, -1,0).

```

Figura 4.2: Creencias iniciales de un agente NEA taxista

Respecto a la parte normativa, en nuestra propuesta hemos comentado que las normas se modelan como percepciones, que después el agente adquiere como creencias. Por lo tanto, en nuestro ejemplo, después de recoger las normas la base de creencias actualizada del agente contendría la norma:

prohibition(respectLine, skipLine, -1, 0).

El taxista puede realizar las siguientes acciones: avanzar en la cola, recoger un cliente y saltarse la cola. Para dar soporte a estas acciones, se han implementado los planes que se muestran en la figura 4.3, donde el plan cuyo evento activador es *spot(N):[priority(1)]* permite al agente moverse dentro de la cola de manera ordenada, i.e. respetando la norma. El plan cuyo evento activador es *takeClient[priority(2)]* permite al agente recoger a un cliente siempre que esté en la primera posición y se encuentre trabajando. Por último, el plan cuyo evento activador es *skipLine[priority(3)]* permite al agente colarse en la fila.

Este último plan se encuentra afectado por la prohibición *respectLine*, por lo que cada vez que sea elegido por la función de selección (que se ha modificado y se explica más adelante), se procede a realizar el razonamiento normativo. En el plan se muestra cómo primero se recogen los parámetros normativos y emocionales necesarios para realizar la función y se procede a realizar el razonamiento normativo (*functions.normativeReasoning(...)*). De manera que la decisión tomada se pasa a la acción *skipLine* que actuará en consecuencia con la decisión del agente.

```

29 // Plans
30 +spot(N)[priority(1)]: state(working) & line(M) & (M-1)==N
31 <- move_towards(0,N);
32   -spot(N);
33   -line(_).
34
35
36 +takeClient[priority(2)]: line(1) & state(working)
37 <- getIn(0,8);
38   -line(_);
39   +state(driving);
40   .wait(200);
41   +line(4);
42   +state(working).
43
44 //Normative plan
45 +skipLine(P)[priority(3)]: spot(N) & state(working) & line(M)
46 <- ?obligation(respectLine,_,_,S,R);
47   ?remorse(RE);
48   ?pride(PR);
49   ?pE(A,B,C);
50   math.div(math.sum(A,B,C,SUM),3,PE);
51   ?ri(RI);
52   functions.normativeReasoning(S, R, RE, PR, PE, RI, WILL);
53   skipLine(0,N,WILL);
54   -spot(N);
55   -line(_).
56

```

Figura 4.3: Planes de un agente NEA taxista

4.2.2. Modificaciones de funciones de Jason

Función de selección (S_E)

La función de selección de eventos que ofrece Jason no es apta para implementar la arquitectura que proponemos, por lo que como se comentó en el capítulo anterior se modifica tal y como se muestra en la Figura 4.4. Para ello se ha customizado la clase Agente y se ha modificado la función de selección de manera que se priorice al evento cuyo plan se encuentra afectado por la norma. Los planes tienen asociados prioridades. Por este motivo, los eventos cuyos planes sean afectados por las normas tendrán siempre una prioridad mayor al resto. De este modo, aplicando esta función de selección, tendrán prioridad y se permite así que el agente razone si, teniendo la opción de violar la norma, lo quiere hacer.

```

10 public class TaxiAgent extends Agent {
11     private Trigger focus = Trigger.parseTrigger("+skipLine(N)[priority(3)]");
12
13     @Override
14     public Event selectEvent(Queue<Event> events){
15         Iterator<Event> i = events.iterator();
16         while(i.hasNext()){
17             Event e = i.next();
18             if(e.getTrigger().equals(focus)){
19                 i.remove();
20                 return e;
21             }
22         }
23         return super.selectEvent(events);
24     }
25 }

```

Figura 4.4: Función de selección de evento de un agente NEA taxista

Función razonamiento normativo (NR)

Esta función se ha modelado dentro de Jason como una acción interna, de manera que el razonamiento normativo puede realizarse dentro de los planes cuyos eventos activadores se encuentren afectados por una norma. Esta acción implementa la función que definimos en el capítulo anterior para realizar el razonamiento normativo, de manera que el valor obtenido en la variable *will* se corresponde con la decisión que tome el agente. En la figura 3.5 se muestra el código asociado.

```
7 public class normativeReasoning extends DefaultInternalAction{
8     @Override
9     public Object execute ( TransitionSystem ts, Unifier un, Term [] args) throws Exception{
10         try{
11             //1. Get the arguments
12             NumberTerm s = (NumberTerm)args[0]; //Sanction (S)
13             NumberTerm r = (NumberTerm)args[1]; //Reward (R)
14
15             NumberTerm rem = (NumberTerm)args[3]; //Emotion remorse (e_{rem})
16             NumberTerm org = (NumberTerm)args[2]; //Emotion pride (e_{org})
17
18             NumberTerm pE = (NumberTerm)args[4]; //Probability of been seen (P_{E})
19             NumberTerm ri = (NumberTerm)args[5]; //Reputation importance (ri)
20
21
22             //2. Calculate will
23             double violation = Math.abs(s.solve()+rem.solve()*pE.solve()*ri.solve()); //parameter Theta_{violate}
24             double fulfill = Math.abs(r.solve()+org.solve()*pE.solve()*ri.solve()); //parameter Theta_{fulfill}
25             double will; //Parameter Theta_{will}
26
27
28             if(fulfill > violation){will=1;} //Fulfill the norm
29             else if (fulfill < violation){will=-1;} //Violate the norm
30             else{will = 0;} //Decide random
31
32
33             NumberTerm result = new NumberTermImpl(will);
34             return un.unifies(result,args[6]);
35
36         }catch (ArrayIndexOutOfBoundsException e){
37             throw new JasonException("The internal action <normativeReasoning> has not recieved 7 arguments!");
38         }catch (ClassCastException e){
39             throw new JasonException("The internal action <normativeReasoning> has recieved arguments that are not numbers!");
40         }catch (Exception e){
41             throw new JasonException("Error in internal action <normativeReasoning>");
42         }
43     }
44 }
```

Figura 4.5: Función que realiza el razonamiento normativo del agente NEA taxista

4.3 Discusión

La implementación se encuentra en proceso aún por lo que la función que actualiza la base de emociones (BEA) y la función que actualiza el estado emocional del agente (EA) no han sido implementadas. Esto y otros aspectos como la ejecución y análisis del caso de estudio con distintos roles de agentes (transgresor, dubitativo, respetuoso) se deja como trabajo futuro.

CAPÍTULO 5

Conclusiones

En este capítulo se recogen las conclusiones obtenidas de este trabajo, presentando las aportaciones que ofrece en su campo de estudio, así como las limitaciones que presenta. También se comentan las dificultades encontradas durante el trabajo, así como las líneas futuras de investigación y el trabajo que queda por realizar.

5.1 Aportaciones y limitaciones

La arquitectura de agentes NEA propuesta en este trabajo modela agentes normativos emocionales que por una parte dan soporte a las normas, sin quitarles su autonomía como muchas propuestas normativas hacían, y por otra parte dan soporte también a las emociones, permitiendo mostrar el efecto que estas tienen sobre las normas y viceversa. Para ello, se ha propuesto en este trabajo:

- Un *modelo normativo* que permite la representación de las normas, que se modelan como parte del entorno de manera que son recogidas como percepciones por parte de los agentes. Así, no hace falta que las normas se introduzcan en ellos directamente, sino que este modelo permite respetar la autonomía de los agentes. La monitorización se lleva a cabo por los mismos agentes del sistema, de manera que el parámetro de *visibilidad* propuesto, permite simular lo que pasaría en un entorno real, donde una transgresión o cumplimiento de una norma puede ser vista o no por el resto de personas del entorno y así queda plasmado en este modelo.
- Un *modelo emocional* basado en la teoría de valoración, más específicamente en la teoría emocional OCC, que hemos simplificado y adaptado para crear un modelo emocional en el que el agente es capaz de experimentar cinco emociones distintas sobre sí mismo y dos emociones anticipadas que el agente vuelca sobre el resto de agentes como respuesta normativa a las acciones que observa en ellos (cumplimiento/transgresión de una norma). Permite también regular la importancia que el agente da a las emociones normativas que recibe, gracias al parámetro *reputation importance*.
- Una función *normative reasoning (NR)* que se encarga de realizar el razonamiento normativo y tiene en cuenta tanto los factores normativos como los emocionales, permitiendo así dar una respuesta más real que la que proporciona un agente solamente racional o solamente normativo.

- Una función *emotion actualization* (EA) que se encarga de actualizar el estado emocional del agente de manera que es capaz de reflejar los efectos que las decisiones normativas (y por ende las normas) tienen sobre su estado emocional.
- Una función *base emotion actualization* (BEA) que se encarga de actualizar la base de emociones del sistema y además se ocupa de manejar la recepción de las repuestas emocionales a las acciones normativas que ha ejecutado el agente.

Aparte de estas aportaciones, cabe destacar que del análisis del estado del arte realizado en el capítulo 2, hemos observado que los efectos entre las normas y emociones pueden resumirse en cuatro: (1) las emociones influyen directamente en el proceso de razonamiento del agente; (2) el cumplir o violar una norma genera emociones en el agente (e_{org}, e_{rem}); (3) las emociones ($e_{gatif}, e_{ad}, e_{rep}, e_{enojo}$) que recibe el agente por parte del resto de agentes como respuesta a sus acciones normativas hacen presión social que puede influir en la decisión del agente de cumplir o no la norma (*Norm enforcement*); y por último (4) las emociones pueden ayudar a inferir normas privadas (ver figura). En lo que a nosotros respecta, ninguna de las propuestas existentes sobre agentes (autónomos) normativo-emocionales permite a los diseñadores modelar agentes que exhiben estas cuatro relaciones entre las normas y emociones que hemos observado, por ello nuestra aportación de un agente NEA, que es capaz de reflejar tres de las cuatro relaciones.

Nuestra propuesta es bastante ambiciosa por lo que en este trabajo no es posible abarcar y dar solución a todos los aspectos que conllevan la implementación de un agente normativo-emocional. Por ello las limitaciones o aspectos no resueltos que presenta nuestra propuesta son: el proceso de resolución de conflictos, el aspecto emocional de la personalidad del agente, así como modelar su estado de ánimo. Tampoco se aborda la cuarta relación entre normas y emociones que implica la generación de normas privadas, así como no se da soporte a la generación y creación de emociones de manera dinámica, de modo que en nuestra propuesta las normas se introducen *offline*. Además la función que realiza el razonamiento normativo no tiene en cuenta el beneficio del plan, como sí lo hace Elster [21].

5.2 Dificultades y trabajo futuro

Este trabajo ha presentado algunas dificultades, especialmente durante el proceso de implementación, que cabe mencionar. En un principio para no partir desde cero, se decidió partir de un modelo normativo ya implementado, sobre el cual incorporar las emociones. Con esta idea en mente después del análisis del estado del arte de sistemas normativos, encontramos que la propuesta de Meneguzzi presentaba un código que podía descargarse y ser manipulado. Sin embargo todo el tiempo y trabajo invertido en este modelo resultó en vano pues por una parte no se consiguió poner en correcto funcionamiento el software y además la representación de las normas que adoptaba no se correspondía con la representación que por mi parte deseaba modelar. Por esto al final se decidió partir de cero y que este trabajo proporcionase el modelo normativo, pensado completamente. Es por esta razón también que en última instancia la parte de la implementación y el caso de estudio que se presentan, están desarrollados pero no completamente y se dejan como trabajo futuro. Así como también los siguientes aspectos que durante el proceso de este trabajo hemos observado que serían ideas interesantes a desarrollar en un futuro:

- Hacer que los valores de las emociones normativas externas orgullo (e_{org}) y reproche (e_{rep}), que en este trabajo son especificadas *offline*, puedan ser valores dinámicos cuyos

valores se encuentren afectados, i.e., se calculen en función del estado emocional actual que presente el agente.

- Incorporar la aportación de las emociones anticipadas enojo y gratificación que en este trabajo sólo hemos citado pero no las hemos empleando, pudiendo en un futuro modelar la emoción anticipada que responderá el agente observador, como un valor que oscile entre un rango y cada rango asociarlo con una de las cuatro emociones anticipadas (del mismo modo que hacemos para el cálculo del estado emocional).
- Omitir la asunción que tomamos al limitar la respuesta emocional normativa de manera que si se trataba de una violación a la norma se respondía con la emoción reproche y si se trataba de un acto de respeto a la norma se respondía con la emoción admiración. Sin embargo es interesante quitar esta limitación y permitir responder con cualquiera de las cuatro emociones que se pueden generar en el agente observador (ver tabla 3.1): gratificación, admiración, reproche y enojo.

Respecto a los desafíos que presenta el estudio de agentes normativos emocionales, a continuación se comentan algunas líneas abiertas en este campo de investigación:

- *Creación de normas privadas (Moralidad)*: como se explicó anteriormente, las normas privadas son aquellas que son autoimpuestas por el agente. Sería interesante que un agente pudiera no solo decidir si sigue o no las normas del sistema, sino también ser proactivo y, al analizar el entorno, podría inferir sus propias normas privadas, utilizando su propia experiencia sobre lo que funciona mejor. Esta experiencia también incluye emociones. Cuando se combinan normas privadas con emociones, entonces pueden considerarse como reglas de moralidad. Pankov y Dastani propusieron una especificación semiformal de tres emociones morales (ira, desprecio y disgusto) [70]. Estas emociones morales deben ser consideradas principalmente por el agente al inferir sus normas privadas.
- *Auto cumplimiento de las normas*: La aplicación de la norma ha sido implementada principalmente por [16]: entidades de segunda parte, donde los agentes directamente involucrados en una interacción se encargan de monitorizar y tomar medidas coercitivas en consecuencia; o entidades de terceros a cargo de aplicar sanciones en caso de violación de la norma. Sin embargo, la auto-aplicación apenas ha sido considerada. Como se explicó anteriormente, las emociones podrían usarse como un mecanismo adecuado para hacer cumplir las normas sin necesidad de una segunda o tercera parte, pero solo el propio juicio personal del agente, modelado por medio de sus propias emociones. Este auto-cumplimiento puede ser aplicado a normas institucionales, convenciones y normas privadas.
- *Personalidad, estado de ánimo y relaciones emocionales*: La personalidad se refiere al conjunto de rasgos individuales que hacen a las personas diferentes entre sí [49, 33]. El estado de ánimo puede considerarse como un estado de ánimo o sentimiento temporal. El estado de ánimo es generalmente más estable que una emoción particular, aunque ciertamente están involucrados [75]. En el campo psicológico, hay muchos trabajos que se relacionan con las emociones, el estado de ánimo y la personalidad. Por ejemplo, Rusting [73] encontró que el humor y la personalidad temporales influyen en el procesamiento de la información emocional en diferentes tareas cognitivas. Gebhard [49] integra las emociones, el estado de ánimo y la personalidad para lograr un procesamiento afectivo en agentes autónomos conversacionales. Por lo tanto, al tomar en cuenta las emociones, la personalidad del agente también está directamente involucrada. Las emociones permiten

flexibilidad en la interpretación y respuesta del evento dependiendo de la personalidad dada al agente. Sin embargo, esta relación emoción-personalidad no se ha abordado todavía para los NEA.

Bibliografía

- [1] Rafael H. Bordini, University of Durham, U., Jomi Fred Hubner, University of Blumenau, B., & Michael Wooldridge, University of Liverpool, U. (2007). programming multi-agent systems in AgentSpeak usinf Jason.
- [2] Criado, N. (2012). Using Norms To Control Open Multi-Agent Systems, (October).
- [3] Garcia, E. (2014). Engineering Regulated Open Multiagent Systems, 417–419. <https://doi.org/10.3233/AIC-140610>.The
- [4] Picard, R. W. (n.d.). Affective Computing, (321).
- [5] J. Carmo and A. J. Jones. Deontic logic and contrary-to-duties. In Handbook of philosophical logic, pages 265–343. Springer, 2002.
- [6] Wooldridge, M. (2009). An Introduction to MultiAgent Systems - Second Edition.
- [7] Meneguzzi, F., and Luck, M. (2009). Norm-Based Behaviour Modification in BDI Agents. 8th International Joint Conference on Autonomous Agents and Multiagent Systems - AAMAS '09, 177–184.
- [8] Neto, B. dos S. (2011). Using jason to develop normative agents. Advances in Artificial ... , 143–152. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-16138-4_15
- [9] Viana, M. L., Alencar, P., Guimarães, E. T., Cunha, F. J. P., Cowan, D., and Lucena, C. J. P. (2015). JSAN: A Framework to Implement Normative Agents, (July), 660–665. <https://doi.org/10.18293/SEKE2015-165>
- [10] Lee J., Padget J., Logan B., Dybalova D., Alechina N. (2014) N-Jason: Run-Time Norm Compliance in AgentSpeak(L). In: Dalpiaz F, Dix J., van Riemsdijk M.B. (eds) Engineering Multi-Agent Systems. EMAS 2014. Lecture Notes in Computer Science, vol 8758. Springer, Cham
- [11] Meneguzzi, F., Rodrigues, O., Oren, N., Vasconcelos, W. W., Luck, M. (2015). BDI reasoning with normative considerations. Engineering Applications of Artificial Intelligence, 43, 127–146. <https://doi.org/10.1016/j.engappai.2015.04.011>
- [12] Shams, Z., De Vos, M., Padget, J., and Vasconcelos, W. W. (2017). Practical Reasoning with Norms for Autonomous Software Agents. Engineering Applications of Artificial Intelligence, 65(March), 388–399. <https://doi.org/10.1016/j.engappai.2017.07.021>
- [13] Etzioni, A.: Normative-affective factors toward a new decision-making model. Journal of Economic Psychology, 9, 125–150 (1988)

- [14] Bagozzi, R. P., Pieters, R.: Goal-directed Emotions. *Cognition & Emotion*. 12(1), 1–26 (1998)
- [15] Posner, E.: The regulation of solidary groups: The influence of legal and nonlegal sanctions on collective action. *University of Chicago Law Review*, 63, 99–133 (1996)
- [16] Criado, N., Argente, E., Botti, V.: Open issues for normative multi-agent systems. *AI communications*, 24(3), 233–264 (2011)
- [17] Boella, G., van der Torre, L., Verhagen, H.: Introduction to the special issue on normative multiagent systems. *Journal of Autonomous Agents and Multi-Agent Systems*, 17, 1–10 (2008)
- [18] Boella, G., van der Torre, L., Verhagen, H.: Introduction to normative multiagent systems. In: *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings (2007)
- [19] von Wright, G.H.: Deontic logic. *Logical Studies*, 58–74 (1957)
- [20] Rubino, R., Sartor, G.: Preface. *Journal of Artificial Intelligence and Law*, 16(1), 1–5 (2008)
- [21] Elster, J.: Social Norms and Economic Theory. *Journal of Economic Perspectives*, 3(4), 99–117 (1989)
- [22] Tuomela, R.: *The Importance of Us*. Stanford University Press (1995)
- [23] Dignum, F.: Autonomous norms. *Journal of Artificial Intelligence and Law*, 7(1), 69–79 (1999)
- [24] G. Boella and L. van Der Torre. Substantive and procedural norms in normative multi-agent systems. *Journal of Applied Logic*, 6(2):152–171, 2008.
- [25] Savarimuthu, B. T. R., and Cranefield, S.. Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems. (2011)
- [26] Peng, Y. Bin, Gao, J., Ai, J. Q., Wang, C. H., and Guo, H. (2008). An extended agent BDI model with norms, policies and contracts. 2008 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008
- [27] Weiss G. “Multiagent systems: a modern approach to distributed artificial intelligence”. En: MIT press (1999)
- [28] Rodriguez, L.F., Ramos, F.: Development of Computational Models of Emotions for Autonomous Agents: a Review. *Cognitive Computation*, 6(3), 351–375 (2014)
- [29] Ortony, A., Clore, G.L., Collins, A.: *The cognitive structure of emotions*. Cambridge University Press (1990)
- [30] Frijda, N. H., Kuipers, P., Ter Schure, E.: Relations among emotion, appraisal, and emotional action readiness. *Journal of Personality and Social Psychology*. 57(2), 212–228 (1989)
- [31] Roseman, I.J., Spindel, M.S., Jose, P.E.: Appraisals of emotion-eliciting events. *Journal of Personality and Social Psychology*. 59(5), 899– 915 (1990)
- [32] Hudlicka, E.: Guidelines for designing computational models of emotions. *International Journal of Synthetic Emotions*. 2(1), 26–79 (2011)
- [33] O. P. John, L. P. Naumann, and C. J. Soto. Paradigm shift to the integrative big five trait taxonomy. *Handbook of personality: Theory and research*, 3(2):114–158, 2008.

- [34] H. H. Chang and M. Tuan Pham. Affect as a decision-making system of the present. *Journal of Consumer Research*, 40(1):42–63, 2012.
- [35] G. F. Loewenstein, E. U. Weber, C. K. Hsee, and N. Welch. Risk as feelings. *Psychological bulletin*, 127(2):267, 2001.
- [36] H.R. Pfister and G. Bohm. The multiplicity of emotions: A framework of emotional functions in decision making. *Judgment and decision making*, 3(1):5, 2008.
- [37] E. Fehr and U. Fischbacher. Social norms and human cooperation. *Trends in cognitive sciences*, 8(4):185–190, 2004.
- [38] S. Poria, E. Cambria, R. Bajpai, and A. Hussain. A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, 37:98–125, 2017.
- [39] J. Bach. *Principles of synthetic intelligence PSI: an architecture of motivated cognition*, volume 4. Oxford University Press, 2009.
- [40] J.F. Bonnefon, A. Shariff, and I. Rahwan. The social dilemma of 3–1576, 2016.
- [41] V. Conitzer, W. SinnottArmstrong, J. S. Borg, Y. Deng, and M. Kramer. Moral decision making frameworks for artificial intelligence. In *AAAI*, pages 4831–4835, 2017.
- [42] Marsella, S., Gratch, J., Petta, P.: Computational models of emotion. In: *Blueprint for affective computing: a source book*. Oxford University Press (2010)
- [43] Lin, J., Spraragen, M., Zyda, M.: Computational models of emotion and cognition. *Advances in Cognitive Systems*. 2, 59–76 (2012)
- [44] Marsella, S. C, Gratch, J.: EMA: a process model of appraisal dynamics. *Cognitive Systems Research*.10(1), 70–90 (2009)
- [45] Russell, J.A.: Core affect and the psychological construction of emotion. *Psychological review*. 110(1), 145 – 172 (2003)
- [46] Russell, J.A., Mehrabian, A.: Evidence for a three-factor theory of emotions. *Journal of research in Personality*. 11(3), 273–294 (1977)
- [47] Mehrabian, A.: Pleasure-arousal-dominance: a general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4), 261–292 (1996)
- [48] Becker-Asano, C., Wachsmuth, I.: Affective computing with primary and secondary emotions in a virtual human. *Autonomous Agents and Multi-Agent Systems*, 20(1), 32–49 (2010)
- [49] Gebhard, P.: ALMA: a layered model of affect. In: *4th International joint conference on autonomous Agents and Multiagent Systems*, pp. 29–36 (2005)
- [50] Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, chpt. 2
- [51] B. Alfonso Espinosa. *Agents with Affective Traits for Decision-Making in Complex Environments*. PhD thesis, Universitat Politecnica de Valencia, 2017.

- [52] Bordini, R. H., Hübner, J. F.: Jason, manual, release 0.7 edition, Aug. 2005. <http://jason.sf.net/>.
- [53] M. Bratman. Intentions, plans, and practical reason. Harvard Univ. Press, Cambridge, 1987.
- [54] Ekman, P.: Basic emotions. In: Handbook of cognition and emotion. Wiley, New Jersey, pp. 45–60 (1999)
- [55] Velasquez, J.D.: Modeling emotions and other motivations in synthetic agents. In: 14th national conference on artificial intelligence and 9th conference on innovative applications of artificial intelligence. AAAI Press, pp. 10–15 (1997)
- [56] Staller, A., Petta, P.: Introducing emotions into the computational study of social norms: A first evaluation. *Journal of artificial societies and social simulation*. 4(1), U27–U60 (2001)
- [57] Ferreira, N., Mascarenhas, S., Paiva, A., Dignum, F., Mc Breen, J., Degens, N., Hofstede, G.: Generating norm-related emotions in virtual agents. In: *Intelligent Virtual Agents*. Springer Berlin/Heidelberg, pp. 97–104 (2012)
- [58] N. Ferreira, S. Mascarenhas, A. Paiva, G. Di Tosto, F. Dignum, J. McBreen, N. Degens, G. J. Hofstede, G. Andrighetto, and R. Conte. An agent model for the appraisal of normative events based in in-group and out-group relations. In AAAI, 2013.
- [59] Joffily, M., Masclat, D., Noussair, C. N., Villeval, M. C.: Emotions, Sanctions and Cooperation. *Southern Economic Journal*. 80(4), 1002–1027 (2014)
- [60] Elster, J.: Rationality and the Emotions. *The Economic Journal*, 106(438), 1386–1397 (1996)
- [61] J. Fix, C. Von Scheve, and D. Moldt. Emotion-based norm enforcement and maintenance in multi-agent systems: foundations and petri net modeling. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 105–107. ACM, 2006.
- [62] Staller, A., Petta, P.: Towards a tractable appraisal-based architecture for situated cognizers. In: *Grounding emotions in adaptive systems, Workshop Notes of 5th Int. Conf. of the Society for Adaptive Behaviour (SAB'98)*, pp. 56–61 (1998)
- [63] Huber, M. J.: JAM: A BDI-theoretic mobile agent architecture. In: *Proc. Third International Conference on Autonomous Agents, Seattle*, pp. 236–243 (1999)
- [64] Bazzan, A. L. C., Adamatti, D. F., Bordini, R. H.: Extending the Computational Study of Social Norms with a Systematic Model of Emotions. In: *Brazilian Symposium on Artificial Intelligence*, pp. 108–117. Springer Berlin Heidelberg (2003)
- [65] Von Scheve, C., Moldt, D., Fix, J., von Luede, R.: My agents love to conform: Norms and emotion in the micro-macro link. *Computational & Mathematical Organization Theory*. 12(2-3), 81–100 (2006)
- [66] Ahmad, A., Ahmad, M. S., Mohd Yusoff, M. Z., Ahmed, M.: Formulating Agent's Emotions in a Normative Environment. *Knowledge Technology*, 82–92 (2012)
- [67] Ahmad, A., Ahmad, M. S., Mohd Yusoff, M. Z., Mustapha, A.: A Novel Framework for Normative Agent-based Systems. *MJCAI* (2009)

- [68] Dias, J., Paiva, A.: Feeling and reasoning: A computational model for emotional characters. *Progress in artificial intelligence*, pp. 127–140 (2005)
- [69] Conte, R., Castelfranchi, C.: Understanding the functions of norms in social groups through simulation. *Artificial Societies: The Computer Simulation of Social Life*, UCL Press, pp. 252–267 (1995)
- [70] Pankov, A., Dastani, M.: Towards a Formal Specification of Moral Emotions. In: 2nd International Workshop on Emotion and Sentiment in Social and Expressive Media: Opportunities and Challenges for Emotion-aware Multiagent Systems, 1351(May), pp. 3–18 (2015)
- [71] N. P. Shaw, A. Stockel, R. W. Orr, T. F. Lidbetter, and R. Cohen. Towards provably moral ai agents in bottom-up learning frameworks. In *AAAI/ACM Conference on Artificial Intelligence, Ethics and Society*, 2018.
- [72] D. Howard and I. Muntean. A minimalist model of the artificial autonomous moral agent (aama). In *SSS16 Symposium Technical Reports*. Association for the Advancement of Artificial Intelligence. AAAI, 2016.
- [73] C. L. Rusting. Personality, mood, and cognitive processing of emotional information: three conceptual frameworks. *Psychological bulletin*, 124(2):165, 1998.
- [74] B. F. Malle, M. Scheutz, T. Arnold, J. Voiklis, and C. Cusimano. Sacrifice one for the good of many?: People apply different moral norms to human and robot agents. In *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*, pages 117–124. ACM, 2015.
- [75] P. C. Schmid and M. S. Mast. Mood effects on emotion recognition. *Motivation and Emotion*, 34(3):288–292, 2010.
- [76] D. Arellano, J. Varona, and F. J. Perales. Generation and visualization of emotional states in virtual characters. *Computer Animation and Virtual Worlds*, 19(3-4):259–270, 2008.
- [77] G. Andrighetto, D. Villatoro, and R. Conte. Norm internalization in artificial societies. *AI Communications*, 23(4):325339, 2010.
- [78] B.T.R. Savarimuthu and S. Cranefield. A categorization of simulation works on norms. In *Proc. of the International Workshop on Normative Multiagent Systems (NORMAS)*, pages 39-58, 2009.