Exploring neglected avenues in the modelling of attribution theory

Arthur Melissen, s9911472

August 25, 2008

Supervisors:
Prof. Dr. Ir. Anton Nijholt
Department of Human Media Interaction, University of Twente
Prof. Jonathan Gratch
Institute for Creative Technology, University of Southern California
dr. Dirk Heylen
Department of Human Media Interaction, University of Twente
dr. Mariët Theune
Department of Human Media Interaction, University of Twente

Summary

This thesis is part of the requirements for completion of a Master's degree in Computer Science at the Department of Human Media Interaction from the University of Twente in the Netherlands. The subject of the thesis concerns work done at the Institute for Creative Technology (ICT) in Marina del Rey in California. The field of research is that of Virtual Humans, sometimes referred to as agents. While staying at the ICT, the assignment was to improve upon the modelling of attribution theory.

In order to solve this problem, a study was made of the field of attribution theory and the computational models that implement it. After analyzing several potential points for improvement, the choice was made to extend the model presented by Mao by taking the aspect of negligence into account. Literature pertaining to negligence was also examined, and a model that allows inference of negligence by agents in a relatively simple agent environment was formed.

Foreword

First and foremost, my acknowledgements to my advisor, professor Jonathan Gratch of the University of Southern California's Institute of Creative Technology Emotions research group. Jon steered me in the right direction, always providing a fresh angle of view on what I was working on. The work described in this thesis could not have happened without his critical view.

Next, I would like to acknowledge the rest of my advisory committee, professor Anton Nijholt, dr. Dirk Heylen and dr. Mariet Theune for their support and patience during the long intervals both while in Los Angeles and back at home when they were eagerly awaiting new results but had to wait a little longer. I am also thankful for their insights into managing the complexities of modelling an impossibly complex system, the human mind.

Living and working in Los Angeles has been a great experience for me. Running, cycling, snowboarding and sunday morning exercise classes all felt great to be a part of. Tuesday and thursday evening was time for volleyball with the rest of the ICT team. Together we had a great summer with lots of laughs in a good sporting atmosphere. The things I miss most about LA are the diversity of food, and the diversity of people. It is a great place I hope some day to return to.

Contents

Introduction

With the increasing complexity of computer systems, the need to provide a more abstract representation of information processes is growing. At the same time, these advances allow the translation process between the fundamental differences in the way humans and computers process information to be offloaded from the human to the computer. In it's most extreme form, this translation process will involve the computer presenting a manifestation that is for all intents and purposes equal to the entity a human is used to communicate with on a daily basis; Another human. These representations are referred to as *Virtual Humans*.

Virtual Humans are logical entities, composed of many different subsystems. There are components for the graphical representation of the human, it's animation, ability to understand natural dialogue, speech, emotions, but also planning and reasoning. The reasoning a Virtual Human does is often based on psychological research, reflecting the way humans themselves tend to reason about their environment.

In this thesis, we will look at one such theory about human social everyday reasoning called *Attribution Theory*. Attribution Theory was first developed by Fritz Heider in 1958 [Heider 1958]. Since then, the theory has been refined considerably, and an attempt has been made to adapt this style of reasoning into a Virtual Human environment [Mao 2006].

We will explain some of the difficulties and points for improvement that can be made in the model as sketched by Mao, paying considerable attention to the way humans reason about negligence. We formalize a model of common sense reasoning about negligence, and present some evidence indicating it's correlation with human data.

The main research question addressed is;
How can we extend the simulation of attribution theory to incorporate a model of negligence and what are the requirements that this model poses on an agent environment?

While Mao's model of attribution theory focuses on causal attributions, it leaves the concept of negligence untouched. In this thesis, we have expanded the model of attribution theory by developing our own framework, specifically designed for attributions of negligence from one agent to another.

This question can be divided into several subquestions, and the thesis is organized so that each chapter should provide the answer to one of the subquestions.

Chapter one will answer the question what attribution theory is and give an introduction to different human perspectives on the subject of negligence. In chapter two, we will take a look at the current state of computational models of attribution theory, chapter three will provide a discussion about possible avenues for improvement. In chapter four we present our model of negligence and show that it corresponds with human intuitions about negligence. In chapter five we develop our agent framework, which allows an agent to reason about negligence and describe some necessary conditions for any agent environment that allows its agents do the same. Chapter six describes some directions for improvement of the model and concludes this thesis.

1 Theoretical background

In the first section of this chapter we will examine attribution theory by highlighting the work of some of its most well-known researchers. The second part will present a more in-depth look at a part of attribution theory; Negligence.

1.1 Attribution theory

When people perceive an event, they ascribe a cause to it. This process is automatic and occurs continuously. When the cause and event are both physically related, for instance the sinking of a rock to the bottom of a pond because of the effect of relative density and the law of gravity, we call this physical causality. When the cause of the effect is attributed to the psychological state of a person or agent, such as that the rock was thrown into the water by a boy because he wanted to impress his friends, then we speak about social causality.

Attribution theory is a psychological theory that deals with social causality. It tries to explain how people come to their conclusion about which cause is related to a certain event.

In this chapter, we shall provide an insight into some of the details of attribution theory and highlight the theories of some of its most well-known researchers; Heider, Kelley, Weiner and Shaver. After this, we shall focus on a component in Shaver's model, negligence, and discuss the current legal and psychological theories surrounding the subject.

1.1.1 Heider

Attribution theory was first developed by Fritz Heider in 1958 [Heider 1958]. In his work, "'The psychology of interpersonal relation"', Heider presents a number of factors that have an impact on the way we perceive another person, and gives us some clues as to how we can deduce certain propositions from a number of observations. In doing so, Heider gives us the first inference rules on which we can base further reasoning regarding the state of an agent.

A clear example of this can be seen by Heider's description of Attribution of Induced Action (ch.9). Heider states that when an agent o is coerced to commit an action x by agent p, we are less likely to attribute responsibility of the outcome of this action to o than if o were to handle voluntarily.

Furthermore, Heider explains how humans attribute events to either internal or external causes. Internal causes are causes within the actor itself, while external or situational causes are causes outside of the actor.

This leads Heider to classify attribution in a number of distinct categories, which Heider terms as stages or levels;

- I - Association: The agent is not causally connected to the event himself, but is associated with the actual cause. Examples are being a supporter of a winning soccer team or being a citizen of a country.
- II - Impersonal Causality: The agent unknowingly caused the effect to happen.
- III - Responsibility: The agent has foreknowledge about the effect his action will bring about, and therefore is attributed more responsibility than an unknowing agent in level II.

- IV - Intention: The agent acted in the knowledge and intention to bring about the effect. Agents in this level are ascribed the most causal responsibility of all levels.
- V - Coercion: While the agent is still a component in the causal structure leading up the event, the agent will not be attributed all responsibility, because he has been coerced by his environment. Responsibility in this case is at least shared between the environment and the agent.

1.1.2 Kelley

Kelley expanded Heider's model of attribution theory by specifying when people assign internal or external causes based on the covariation principle.

The covariation principle maintains that there usually is a tightly-coupled temporal relationship between the cause and effect. Attribution to internal or external factors is then determined by three factors;

- consistency
- distinctiveness
- consensus

Consistency is the measure in which an agent will, given a specific stimulus, always respond in the same way. Distinctiveness is measured by the way in which the agent responds to the stimulus differs from the way the agent responds to a different stimulus. Consensus is specified by the degree in which other actors will react to the stimulus in the same way.

When all of these factors are present, actors will likely attribute the event to an internal cause. As they diminish, the event is more likely to be attributed to the situation.

1.1.3 Weiner

Weiner's model is an extension to that of Kelley's, where the internal and external causes are referred to as the *locus of causality*. Weiner also adds two extra dimensions called stability and controllability.

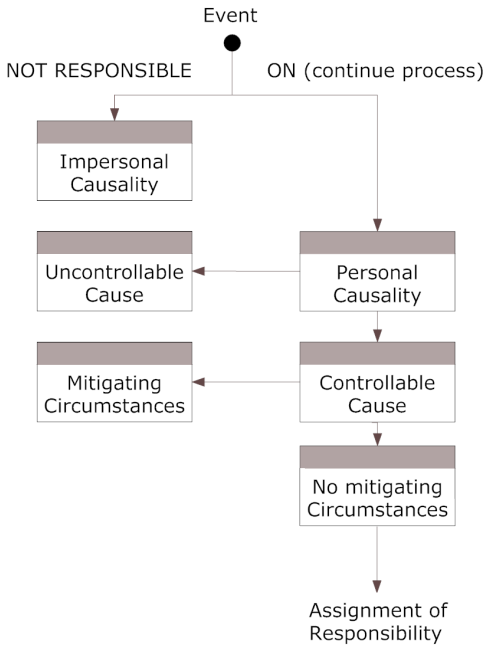
Stability is the perceiver's notion of whether a cause will remain the same for an extended period of time (such as gravity), or whether it will be subject to fluctuations (the weather, for example). Controllability signifies whether the cause is under control of the observer. Both controllable and uncontrollable causes can be internal and external. Archetypical examples of internal controllable and uncontrollable causes are effort and ability, while external controllable and uncontrollable causes are outside assistance and luck, respectively.

Weiner argues that attribution is not a 'cold' process, but generates strong emotions, and that the dimensions of stability and controllability of a cause help explain which emotions are being generated. These emotions are then used to determine the resultant action.

For instance, an act committed because of a controlable cause is likely to elicit anger in an observer, while if the cause is uncontrollable, however, the generated emotion in the observer will be sympathy or compassion.

He also argues that mitigating circumstances play an important role in this attribution. Weiner then uses these dimensions to formalize a model of how people attribute responsibility and blame, and how mitigating circumstances effect this judgment of responsibility.

This process is given in the diagram below, taken from Weiner [1995], p. 12.



Interestingly, Weiner argues that responsibility is lessened if the act was the result of negligence rather than an intentional action (Weiner 1995, page 13).

1.1.4 Shaver

Shaver's attribution theory extends that of Heider and adds the dimensions of foreseeability and coercion.

Foreseeability is defined as an agent being able to possess foreknowledge that allow the agent to judge whether a specific effect will occur when an action is executed. Shaver argues that, even though an agent caused an effect, the agent cannot be held responsible for it if the agent did not know the effect was going to occur.

Coercion is desribed as an agent x acting on behalf of agent y, either by request or command. In such cases, Shaver argues that we do not hold x responsible for actions that x didn't intend himself. Instead, responsibility is relayed to y. This process is recursive, and can be iterated until the agent that gave the original order or request is found.

Shaver refers to the internal and external classification scheme as whether there is human agency. Human agency is present whenever an agent is the cause of an effect. Shaver also notes that this can include effects that have occurred because the agent *failed* to act. Human agency is a necessary but not sufficient condition for responsibility, which is in turn necessary but not sufficient for blameworthiness.

Similar to Weiner, Shaver presents a sequential model for determining the degree to which an agent can be attributed as the cause of an effect, be held responsible for it, and ultimately, when blame can be assigned to that agent.

Blame can be avoided or mitigated by providing a justification or excuse. Whether or not this is succesful depends on the acceptance of the excuse by the party that holds the agent responsible.

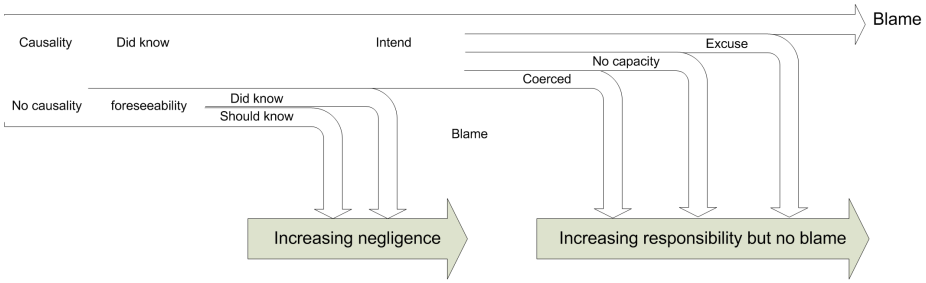


Diagram adapted from [Shaver 1975], p. 166

1.2 Negligence

As mentioned, Shaver presents *human agency* in his model as an agent being potentially responsible for actions that the agent committed or for events that were the result of *failure* of an agent to act. The latter is commonly referred to as negligence, and we will discuss the phenomenon in closer detail in this section.

We will start by looking at negligence from a legal and economical perspective, then examine some of the emotional consequences of when a person is judged negligent, and end by summarizing negligence and its role in attribution theory.

1.2.1 Legal Negligence

In the legal domain, negligence is an important concept, allowing people to be held responsible for not acting in certain situations where a person can be expected to have acted in order to prevent something bad from happening.

The definition of negligence in this domain is 'The failure to use reasonable care'. While it is usually a straightforward process of finding out whether or not a person did or did not commit an action, it is much harder to assess what a person should have done in the given situation and how responsible this person is for the outcome.

In American Law, tackling this problem is done using five stages;

- Duty of care
- Breach of duty
- Factual causation
- Legal causation

- Damage

We will now discuss each of these five stages.

Duty of care The duty of care or standard of care is used to determine the level of care or precaution that the defendant should have taken in order to minimize expected damages suffered by the plaintiff.

The standard of care is highly context-sensitive, and in many cases preset community standards of care based on the relationship between the defendant and plaintiff exist. Examples of this include teacher/student, doctor/patient or accountant/client relationships.

While the accountant will not be held liable due to medical costs associated with an undiagnosed ailment, the doctor is *expected* to diagnose and treat his patient. Failure to do so will expose the doctor but not the accountant to liability.

When there is no community standard of care, the courts may refer to the Hand Rule [Cooter 1991, Feldman 1998]. The Hand Rule or 'calculus of negligence' was first introduced by Judge Billings Learned Hand in the case of *United States v. Carroll Towing*, 159 F.2d 169 (2d Cir. 1947).

The Rule states that when the cost of precaution for preventing an accident or burden, B, is less than the probability p of the accident occurring multiplied by the expected loss L, the party is deemed negligent.

$$B < p * L$$

The case in fact was about a barge that was secured improperly and consequently broke away. The barge drifted off and collided with a tanker. Not only was the tanker's cargo damaged, the barge had sunk.

Judge Hand used the now famous Hand Rule for deciding that the company that owned the barge, Connors Marine Co., was negligent because it failed to have a bargee on board at the time of the accident. Hand's reasoning was that the cost of having a bargee on board would have resulted in some damages, but not in sinking and the loss of cargo. The company was negligent because the personnel costs of having the bargee onboard were less than the expected costs of the accident.

Breach of duty There is a breach of duty when it can be proven that the defendant did not apply a duty of care towards the plaintiff. This means that the harm must be reasonably foreseeable, the defendant did not appropriately respond to address the issue, and demanding a compensation for the resulting damage is deemed fair.

Factual causation It must also be proven that the negligence in fact caused the damage, so that the damages would not have occurred had the defendant not been negligent. This is a hard point to prove, because it involves counterfactual reasoning in hindsight.

Legal causation Legal causation, also known as remoteness, is similar but distinct from factual causation. While factual causation concerns itself with the question of whether or not the negligence is the cause of occurred harm, legal

causation addresses the question of whether the defendant can reasonably be expected to have anticipated this effect given the circumstances. The purpose, therefore, of legal causation is to put a limit on what people can be held accountable for. If no-one could have anticipated the effects of something, than nobody can be held responsible for it.

Damage Once it is determined the defendant is legally responsible for negligent behavior, the only remaining point of interest is to determine how much the victim of this negligence should be compensated for.

1.2.2 Economical models of negligence

Since it's conception, the Hand Rule has been used as a rule of thumb in the courts, but has also sparked considerable research interest in the field of economics. The first economic evaluation of it has been made by J.P.Brown [Brown 1973]. Brown formalizes the notion of costs in terms of the *social optimum*. This social optimum is defined as the minimum of total costs for all parties involved, which consist of the precaution costs for both parties to prevent and mitigate the accident, and the accident costs themselves in case of injury.

The social optimum is influenced by the application of *liability rules*, which determine the distribution of costs given the level of precaution exerted by both the defendant and plaintiff. Examples of liability rules are *no liability*, in which the plaintiff pays for all costs, *strict liability*, where the defendant is responsible for all costs, and *the negligence rule*, where the defendant pays only when he is found negligent, and the plaintiff otherwise. This list of examples is by no means exhaustive but should provide an idea of how the application of a given liability rule influences decisions made by injurers to take a certain level of precaution.

Brown argues that the rule proposed by Judge Hand as a standard of care is ambiguous and uses game theory to analyze a combination of different interpretations of the Hand Rule and liability rules to see which if any will provide an equilibrium that is equal to the social optimum. Combinations that gravitate towards the social optimum are defined as *efficient*.

In Brown's model, precaution exercised by either party is represented as a continuous variable. Attribution of negligence is seen as a two-step process, where first the standard of care defines a minimum amount of precaution, and the second step is to determine whether both parties adhered to this standard.

In more recent work, however, Brown's model has been criticized, argueing that it doesn't represent the way in which the courts apportion negligence [Grady 1983]. Instead, Grady argues that the courts use a *cost-benefit analysis* model, where they attribute negligence by looking for specific precautions that the defendant could have taken but didn't [Grady 1989]. Precautions that would have benefits outweighing the costs of implementing them could be used directly as proof that the negligent party did not observe the standard of care.

Feldman *et al.* [Feldman 1998] support this notion by argueing that many precautions are not continuous by nature but discrete, and that the assumption of continuously variable levels of precaution models some liability rules as efficient which are not efficient under a discrete model.

Recent work by Gilles [Gilles 2002] suggests that in England, the actual practice of the courts is a mixture of a variation on the Hand Rule and cost-benefit analysis.

1.2.3 Emotions in negligence judgements

While there is certainly a wealth of research done on the economic and legal point of view of negligence attribution, as well as the emotional arousal in causality, research that focuses on the emotional arousal of negligence attribution is almost non-existant.

Part of this can be explained by assuming that people do not differentiate emotionally between harm done by causality and harm done because of negligence. In that case, the body of literature pertaining to emotional arousal in causality or general attribution theory can be directly utilised to predict emotional arousal in attributed negligence.

Some of the only work that does target the relationship between negligence and emotion directly is that of Feigenson, Salovey and Park [Feigenson 2001].

In their research, jurors exhibited a strong correlation between emotional arousal and the allocation of blame towards the defendant or plaintiff. While sympathy is generated for the plaintiff who could not be held responsible for any part of the negligent action which resulted in the sustained injury, anger was the primary emotion targeted towards the defendant who did not apply the appropriate measure of due care. Their work also showed a correlation between ambiguity of the allocation of blame and the magnitude of generated emotions.

An explanation that attribution theory can offer us is that negligence can be seen as a having an internal locus of control on the part of the negligent actor, and an external locus in the neglected actor. Thus, the negligent party caused harm through a controllable cause, and elicited anger in the observer. The neglected party suffered from harm through an uncontrollable cause, and consequently aroused sympathy in the observer.

1.3 Discussion

While Heider originally characterized attribution theory as 'common-sense', today it has expanded into an entire field of research and is able to explain a wide spectrum of human behavior, from how to let students perform better academically [Noel 1987] to getting children to stop playing with magic markers [Lepper 1973].

While the attention to the role of negligence is overshadowed by that to the role of causation in attribution theory, negligence still plays an important role in people's everyday judgments of responsibility. We can easily observe this when imagining the placid bystander of a traffic accident or hear about one of the many court-cases involving medical personnel not maintaining the standard of due care.

In correspondence with general intuition and the works of Heider and Weiner, Feigenson *et al.* show us that these scenarios often involve strong emotions, and that they play a critical part in accurately modelling an actor's assessment of blame.

2 Computational frameworks

In recent times, there have been some efforts to formalize attribution theory into computational models. One of the most daunting challenges of converting psychological theory into a computational model is to find a formal representation of the way an agent views the world. This is also called an agent model, and one of the most succesful templates is Bratman's [Bratman 1987].

First, we will discuss some efforts concerning formalization of attribution theory, most notably Mao's and Tomai-Forbus's. We will end by discussing the differences between these models.

2.1 Mao

Mao has developed a computational framework that establishes a foundation for the way agents can reason about social causality. Based on both Weiner and Shaver's attribution theories, it lays the groundwork for an architecture which is able to capture basic human notions of causality.

Agents follow Bratman's BDI-model, and are able to make inferences about causality based on the observation of actions of other agents, as well as the information obtained from dialogues.

The following diagram gives an overview of how Mao's model works.

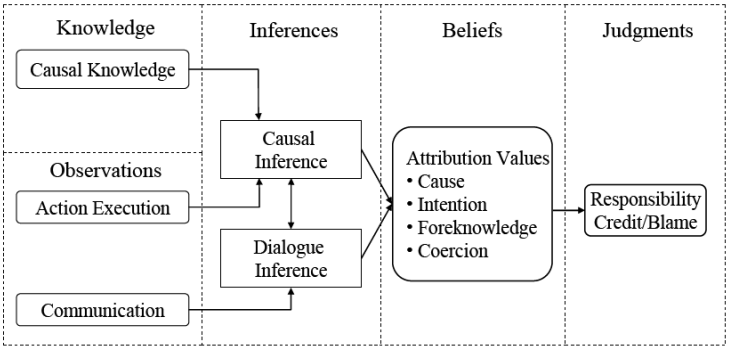


Diagram from [Mao 2006], ch. 3, p. 21

2.1.1 Causal knowledge

The causal knowledge component specifies how an agent receives its information regarding the way in which world state can be altered through the execution of actions. In Mao's model, this information is presented through a plan structure.

A plan structure is a hierarchy of plans, where each plan consists of actions. Actions modify perceived world state by bringing about an effect, and optionally have preconditions. As such, a plan can be represented as a graph, where actions or plans are represented through edges and nodes are places where decisions can be made which course of action to take. A node that has more than one outgoing edge is called a decision-node or abstract node. Contrary to this, nodes that have only one edge are called non-decision-nodes or primitive nodes.

A plan that contains abstract nodes is called an abstract plan. Plans that do not contain any abstract nodes are called primitive plans. Each action in the plan has both a performer and an authority assigned to it. The performer

is the agent who can actually execute the plan, while the authority is the agent who has to authorize the action.

2.1.2 Observations

The agent is able to make observations about the world in two ways. The first is action execution, where the agent observes another agent executing an action. The second is communication, which encodes information taken from dialogue with and between other agents.

Mao represents communication using the form of speech acts as defined by Austin [Austin 1962]. Austin differentiates between three different kinds of communication; illocutionary, locutionary and perlocutionary.

Locutionary speech acts are used to make some general statement about world state *of* saying something (e.g. "The sun is shining"). Illocutionary speech acts are used to encode an agent's desires, knowledge and intentions *in* saying something. Examples of illocution are asking a question or giving an order. Finally, perlocutionary speech acts are intended to change the psychological state of the hearer *by* saying something. Mao focuses mostly on illocutionary speech acts in her work.

2.1.3 Inferences

Using the observed actions and speech acts, an agent in Mao's model is able to combine this with the causal knowledge by using a set of rules to derive a set of attribution variables. These variables contain information about the causal circumstances of an agent, such as whether an agent caused or intended an action, if it had foreknowledge of the consequences and if it was coerced to take action.

2.1.4 Attribution Process

In the second step of Mao's algorithm, she uses the agents beliefs about the attribution variables to come to a conclusion regarding who the primary responsible agent was, and if there were any secondary responsible agents. The primary responsible agent is by default the agent who brought about or caused a consequence. This agent is also known as the *performer*. If the agent was handling in the face of coercion, the *performer* is given secondary or partial responsibility and the coercing agent is given primary responsibility. Because coercing agents can themselves be coerced too, this is a recursive step. Attribution variables *intention* and *foreknowledge* are then used to determine the degree of responsibility. An agent that intended a consequence to happen is given a *high* degree of responsibility, while an agent that had no foreknowledge of the consequence is given *low* responsibility.

2.1.5 Evaluation

Mao's model presents a simple but effective model for attributing blame in an agent environment. Corresponding to the psychological literature, the presence of physical causality, intention and foreknowledge determine for a large part whether or not an agent is found guilty. The addition of coercion makes the model a lot more interesting, creating a differentiation between primary and

secondary responsibility. Mao does not present a way to unify these two concepts along one dimension of responsibility, but leaves them as two related, but different entities. This forms a contrast with the models of Tomai and Forbus, as we'll see in the next section.

2.2 Tomai-Forbus

The second implementation we will discuss in this chapter is that of Tomai-Forbus [Tomai 2007]. Tomai and Forbus used Qualitative Process Theory [Forbus 1984] to step away from Mao's discrete assigment of blame and create a continuous assignment of responsibility across multiple agents.

First, we will provide a summary of Qualitative Reasoning and Qualitative Process Theory, then we will discuss the way in which Tomai and Forbus have implemented causal reasoning in their Qualitative framework, and we will end by providing an insight into the similarities and differences their results when compared with Mao's model.

2.2.1 Qualitative Reasoning

Qualitative Process Theory (QPT) is a theory originally developed to facilitate common-sense reasoning about physical processes. QPT aims to describe physical processes on a conceptual level. Systems are decomposed into collections of objects, the relations between the different objects are the most important part of the system. They allow the system to reason about the effects of various processes throughout the entire system.

While originally designed to handle especially physical characteristics of objects, such as pressure and temperature, exerted force, researchers have found QPT to be suitable to a much wider domain of problems, including education [Bredeweg 2004] and medicine [Fink 1996].

2.2.2 The Qualitative model

The Qualitative model as presented by Tomai and Forbus is based on that of Mao. The components in Mao's model that make inferences from the dialogue and action execution of other agents are left intact. Tomai and Forbus then use these variables as inputs into their qualitative model of attribution theory.

The functionality in their model is distinct from that of Mao's components of beliefs, which are modelled by logical expressions that produce a boolean outcome. Tomai-Forbus' model allows continuous output results, allowing for a much more fine-grained result. The authors have some evidence to suggest this more closely resembles human attribution.

A judgment of causality is divided into four categories, ordered by increasing amount of responsibility; Causality without foreknowledge, causality without intent, coerced causality and causality with intention.

To differentiate between degrees of responsibility within each classification, or views as they are called in their model, they use continuous variables from the qualitative model, such as intention and foreknowledge.

Similar to Mao, Tomai and Forbus hold an agent with authority over another agent responsible for the actions of the inferior agent, if the agent was ordered to commit an action by his superior. In contrast, Tomai-Forbus also present

inference rules which allow the superior agent to be held responsible for not avoiding a bad outcome by instructing the inferior not to do something. This lack of coercion to prevent will result in a lesser degree of responsibility than coerced action to produce a bad outcome.

2.2.3 Evaluation

Evaluation is done by comparing the results of the qualitative model to those of Mao and of human respondents as presented in an experiment held by Knobe [Knobe 2003]. The experiment is called the 'company program'. In the company program, two agents, a vice president (vp) and chairman (chm) of a company, are discussing implementing a new program for the company. Implementation of this new program will bring harm to the environment.

Four variations are presented, in which the chairman and vice president vary according to their foreknowledge and concern about the harm to the environment. Respondents are asked to rate blameworthiness for implementing the new program on a six-point scale. These results are then compared to the Qualitative model and Mao's model, which asserts only a binary answer to the question of blameworthiness.

2.2.4 Results

In all scenarios, the human respondents attributed blame to both agents. The highest amounts of blame were assigned to coercing agents, and agents that had foreknowledge of the environmental harm but proceeded to go ahead anyway.

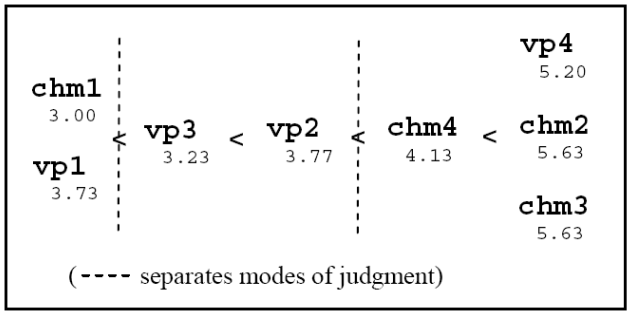


Diagram from [Tomai 2007]

Mao's model correctly locates the agent with the highest amount of responsibility, but fails to reveal a more fine-grained distribution of blame allocated to each agent. The Qualitative model, on the other hand, is able to allocate blameworthiness to both agents, and correctly infers the ordering of responsibility in most cases, even across scenarios.

Interestingly, humans attributed more blame to the agent that had no foreknowledge of environmental harm than to the agent that had foreknowledge but was coerced. This is a violation of Shaver's underlying theory, which classifies an agent with foreknowledge as having more responsibility than one without.

2.3 Discussion

Several attempts have been made to describe attribution theory in an agent model. Mao's agent model closely resembles Shaver's psychological one, in that attribution is a staged process; A set of factors are taken into account sequentially, such as coercion, foreknowledge and intent. Every next step in the model results in increasing responsibility and ultimately blame.

One of the most obvious shortcomings in this approach is the almost binary selections Mao handles for her assignment of responsibility and blame. Tomai and Forbus recognize this, and take a different approach, trying to quantify these along one common dimension but in a finer resolution.

In their work lies the hint of a completely overlooked area of interest: They put responsibility on an authoritative figure to instruct its subordinates not to cause harm, introducing the first mechanism an agent can be held liable without having executed any action, effectively blaming an agent for negligence. In the next chapter, we shall revisit Mao's model and highlight a few points where it can be improved.

3 Model evaluation

In this chapter we will discuss some of the points for improvement for Mao's model.

3.1 Coercion

In Mao's work, two notions of coercion are presented. One is a strict notion of coercion as an absolute quantity, whereby the coercing agent is seen as the primary responsible, and the coerced agent as secondary responsible. The other is a more fine-grained scale, where coercion is measured as a normalized difference in utility between different coerced outcomes. In order to understand the differences between these approaches, we will have to take a closer look at the mechanics of coercion attribution.

A coerced action can consist of an abstract plan, which can be decomposed into a set of primitive plans. A primitive plan has a set of action effects. The set of action effects that are present in the outcome of every primitive plan are called the *definite effect set*. Since a coerced agent has no way of avoiding these effects, coercion is attributed to the entire effect set, mitigating responsibility for the executing agent.

The rest of the effects comprise the *indefinite effect set*. These can be avoided, depending on which decomposition the executing agent chooses, and therefore the executing agent is still seen as responsible for these outcomes.

In the probabilistic model, Mao argues that an agent that has to decompose a coerced abstract plan will likely choose the primitive plan with the highest aggregate utility. Any negative effects in this plan are therefore necessary evils, and attribution of coercion in the probabilistic model is modified accordingly. By looking at the range of expected utilities among the different primitive plans available, an agent that chooses the primitive plan with the highest expected utility among all possible coerced plans should not be blamed for causing a negative effect.

The calculation of degree of coercion is done by way of linearly scaling coercion from zero to one between the minimum and maximum expected utility of all primitive plans. For example, an agent that can choose between three plans with expected utilities of zero, six or ten, and that chooses the plan with expected utility six will receive a degree of coercion equal to 0.6. Another agent that chooses the plan with expected utility ten will be attributed a degree of coercion equal to one, for he has tried to maximize utility under coerced circumstances.

Mao presents results of experiments using human subjects that suggest her model is correct in assigning responsibility to a coercer or superior, but that still could be improved. Evidence of this can be obtained from the fact that the human subjects in the firing squad attributed less coercion than the model did.

We argue that a possible explanation for this is that the current model of coercion fails to capture an essential facet of the way humans attribute coercion to action outcomes; The choice of not obeying the coercer. In the current model, coercion is seen as a binary force applied to an agent resulting in that agent having a reduced choice of options. We argue that humans attribute coercion by reasoning counterfactually about the consequences of not obeying the coercer.

In order to achieve this in the current model, the search for primitive plans with a higher expected utility will have to be expanded to include plans outside of the coerced action. When examining any of these plans, one will have to take into account what the consequences of disobedience are. This is certainly a hard problem, but could be modelled as simply as a negative expected utility. The expected utility of these plans is then decreased by the utility that the agent expects to lose by not obeying a coercer.

The severity of disobeying a coercer can vary greatly. Therefore, the expected utility loss is highly context dependant. Disobeying a teacher's request to do one's homework will have less negative utility than to disobey a direct order from a general in wartime. This last point can be applied more generally, and is the next subject of discussion.

3.2 Social norms

Social norms are not modelled and play an important role in many situations. We can easily observe this when we take a closer look at the phenomenon of coercion: When a child does a chore because it has been commanded to by a parent, we can see this as a soft form of coercion. The situation is very different however, when we look at coercion in a military sense: Disregarding the intentions of a superior officer in the armed forces will have a much stronger influence on one's life, and the degree to which coercion is being effected should be accounted for accordingly.

Likewise, we can look at the question of foreknowledge. In Mao's model, an agent can avoid having high responsibility by not knowing the consequences of its actions. Regardless of whether such a strategy is a useful one in practice, Mao's (and Shaver's) choice to let lack of knowledge form a reason for reduced responsibility is a good one, as people often forgive each other for harm caused when they were not aware of the consequences of their actions.

Nevertheless, situations in which we do hold each other responsible regardless of demonstrated foreknowledge exist in our daily lives too, the legal system being a very good example. Other examples where one is expected to know the consequences of its actions are plenty in professions where there is a risk of harming a client: We expect our doctors to know the consequences of performing a treatment on us, mush as we expect a dive instructor to know the risks associated with diving in unclear waters.

Clearly, we live in a world where not knowing is not always a free pass out of responsibility. The circumstances under which one can, and one can not be expected to be mindful of its actions are not clearcut, however, and we cannot hold Mao responsible for not attempting to integrate the aspect of social norms in her causal framework.

3.3 Negligence

From Shaver we have learned that humans attribute responsibility not only when an actor causes an effect, but also when an actor *ought to* have caused an effect. Shaver describes this as negligence, but does not elaborate much and instead focuses like most attributional theorists on cause by causation, not on the lack thereof.

The impact of this is visible in the attributional model in Mao's work. In the current model, this aspect of negligent agents is not taken into account: An agent that does not perform any actions cannot be blamed for anything in the attributional model as defined by Mao.

Therefore, an optimum strategy for an agent that does not wish to be held blameworthy can be to not execute any actions, and remain a passive observer. The critique here is twofold: First, this is not the way people attribute responsibility in reality, since humans attribute negligence often and swift. Examples are legion and easy to come by; Think for instance of the child that did not clean its room or the passive observer after a traffic accident has occurred.

Second, this is also not what one would generally desire from an agent (or a human): While its a very easy strategy to let an agent choose its actions conservatively and only execute when there is a near certainty of success in order to limit potential blameworthyness, a more admirable approach would probably be to judge an agent on its *potential* actions and blame them for missed opportunities.

3.4 Discussion

Mao's model makes a good approximation of attribution theory in general, allowing an agent to draw inferences not only from the actions of another agent, but also its dialogue with other agents. The modelling of both dialogue and actions is essential, because humans use these two facets of reasoning as well, and without either one, any serious attempt at modelling attribution theory would be seriously handicapped.

However, human notions of attribution are complex, and as we can see from the difference in opinions between psychologists who have pursued to study this field for decades, even things that seem a priori to be clear cut, like intention, turn out to be full of subtle issues when one tries to put them into a model.

In our opinion, the biggest hole to be filled here is that of negligence, since a passive agent in Mao's model can never be punished. A model that ameliorates these difficulties is the subject of the next chapter.

4 A model of negligence attribution

After having discussed some of the points where Mao's model can be improved, we have chosen to model the attribution of negligence. Because of the inherent differences in the type of reasoning that people do when compared to causality attribution, we have chosen to build our own model. This model is presented here.

4.1 Introduction

First, an overview of our framework and its origins is presented. To validate our claims, this model is tested against human notions of negligence attribution. After discussion of the results we will adapt our model to better match the human data.

4.2 Framework overview

For the design of our negligence framework, we start by looking at the way Shaver has modelled negligence in his theory of attribution, and extend it by looking at other factors used in the causal section of attribution theory, like *intention* and *excuses*. Also, we incorporate theory from the legal approach to how negligence is determined, and merge this into a framework similar to the way causal responsibility is established.

The most obvious resemblance to the causal framework as developed by Mao, is the way an agent starts reasoning when a negative consequence of an action is perceived; A number of discrete steps are taken, acting as filters. Finally a judgment of responsibility & blame, or in this case, negligence, is made. Because utility of effects can vary per agent, we define a *negative outcome* or *negative consequence* as an effect from an action that is perceived by at least one agent as being unwanted or having *negative utility*.

By starting with Shaver's view of negligence, an agent is attributed increasing negligence when the agent has foreknowledge of a negative outcome. Shaver divides foreknowledge into two categories: 'should know' and 'did know'. We interpret these two conditions as follows; When an agent knows a negative outcome is about to occur because of an action, this outcome can only be avoided by direct action of the agent. In the case of 'should know', however, the negative outcome may also be avoided by different means outside of the direct influence of the agent, but the agent is also capable of blocking the action.

We keep these two aspects, and extend them with that of *intention*; Analogous to the attribution of responsibility in causal attribution, an agent that intends a bad outcome for another agent to happen is likewise attributed more negligence than an agent that did not intend it.

The next factor we take into account in our model, is that of *possibility*. This entails whether an agent was able to prevent or could know about the negative outcome beforehand. An agent that could not have known about or prevented a negative outcome from occurring is aquitted from negligence.

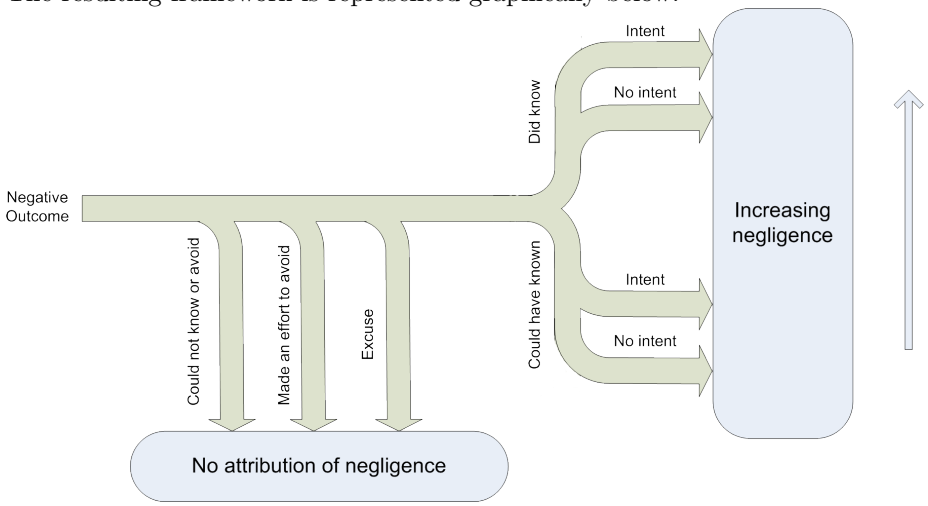
Excuses are another factor we borrow from Shaver's model of causal attribution. They play a prominent role in our framework, because we believe they play a significant role in our everyday attribution of negligence as well. In our model, an excuse is something an agent was doing while he could also have tried

to avoid the negative outcome. A good excuse is one with a greater amount of potential utility than the negative outcome. A bad excuse contains less potential utility than the negative outcome.

Even more inspiration can be drawn from legal studies. The subject of negligence attribution and compensation for the resultant damages is an important and much studied subject, and is the source of a wide variety of economical models which strive to find a social optimum [Brown 1973]. Usually these models focus the attribution of negligence on whether the agent has expended enough *effort* to remove or reduce the risk of adverse effects. This is the last factor we incorporate into our model.

Negligence can also be attributed when there is no bad outcome, but there was a risk of something bad happening. These cases can be described as *unrealized* negligence. Cases where the bad outcome is established can then be referred to as *realized* negligence. In this thesis, we are only concerned with realized negligence.

The resulting framework is represented graphically below:



It is important to note that this does not represent a decision-scheme, where each decision is a binary true or false value. Rather, during each filter stage, it is possible to apportion a ratio of the negative outcome to one of the filters, allowing the rest to pass through to subsequent filters. This creates the possibility of an outcome being attributed partially to different factors, allowing for more fine-tuned control in the form of mitigation. A more detailed description of each of these stages will be given later in this chapter.

4.2.1 Emotional impact

Similar to Weiner's theory of attribution, we expect attributions of negligence to generate emotions in the person who is being negligent, as well as those who suffer the consequences from it. To describe the emotional state of a person, Weiner uses seven emotions: pride, anger, pity, guilt, shame, gratitude and hopelessness.

While we feel that the emotional impact of negligence attribution is certainly related to that of causal attribution, we have opted for a different approach to quantifying the emotional state of an actor. Starting with a very limited set

of 'basic emotions' [Ortony 1990]: we take sadness, happiness, anger and fear. From Weiner's model we then add shame, and guilt. The inclusion of both these emotions is interesting because, although closely related, they are generated under different circumstances, depending on how an event is appraised [Tracy 2006]. Guilt is associated with unstable and controllable causes, whereas shame is associated with stable and uncontrollable causes. We have chosen to model the intention of the agent as a stable, internal and controllable cause, while effort corresponds with an unstable, internal and controllable cause.

The next emotion we take into consideration is sympathy; According to Feigenson *et al.* [Feigenson 1997], people react with sympathy to a victim that is not responsible for it's own suffering, and with anger to a victim that is responsible. Another advantage of sympathy is that it is less ambiguous than pity, which can be classified into two different strands of emotion; benevolent and condescending pity. To avoid questionnaire participant fatigue, we have limited the number of emotions in this research to seven.

Due to the limited quantity of previous research on emotional arousal concerning negligence attribution, the basic emotions allow us to get a general feeling for arousal, while building on theory from Weiner and Feigenson et al. According to Ortony [Ortony 2001], something good happening can lead to joy or happiness, while something bad happening can lead to sadness or distress. We use this and model emotional arousal after an outcome as the result of negligence as follows;

Emotion	Trigger / Conditions
Sadness	The negligent agent suffers from the outcome ($utility < 0$)
Happiness	The negligent agent benefits from the outcome ($utility > 0$)
Anger	The agent perceives another agent's negligence
Guilt	Another agent suffers from the agent's negligence (unstable cause)
Sympathy	Another agent is perceived as suffering (from negligence)
Shame	Another agent suffers from the agent's negligence (stable cause)
Fear	Another agent suffers from the agent's negligence

4.3 Evaluation

After having decided on the factors we take into account in our model and the expected emotional arousal, we must now attempt to validate our assumptions by comparing them to human notions of negligence.

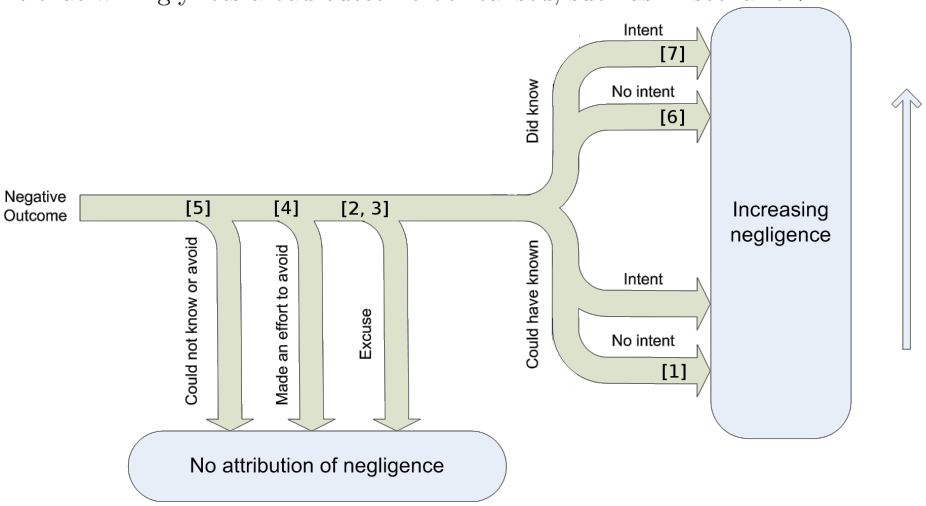
4.3.1 Method

To assess consistency between human reasoning and our framework we have designed a questionnaire named The Goldfish. The Goldfish features a scenario where two people, Andre and Bob, share an apartment. The scenario is given in seven different variations, representing different decision points in the framework. Sentences in the varying part of the scenario are numbered.

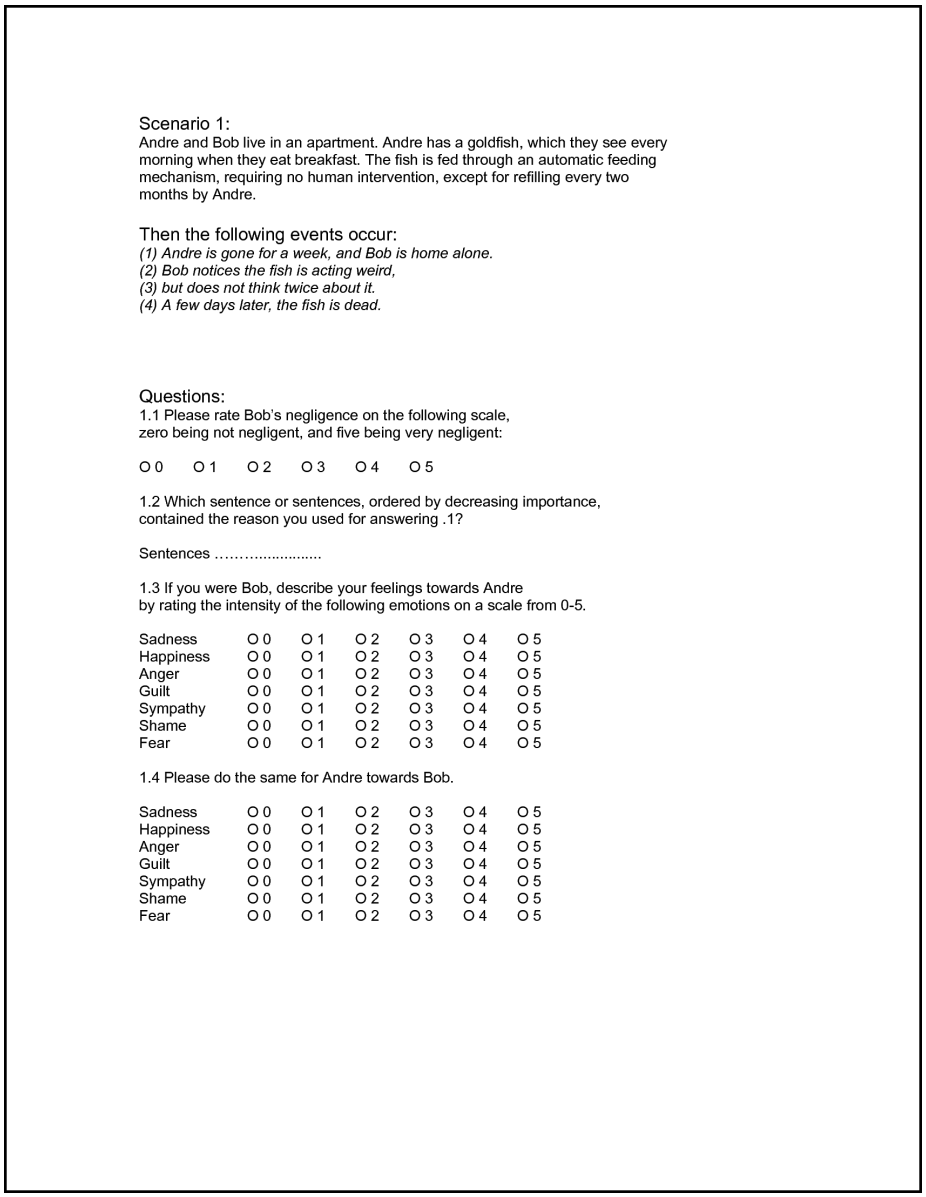
Users are asked to rate the amount of negligence, and to answer which of these sentences contained the most important information to make this judgment on. Also, the user is questioned for the expected arousal of seven different emotions in both of the characters towards each other.

By varying the circumstances under which certain events occur, we can get an assessment of how people reason about negligence when a bad outcome is realized. To detect possible ordering effects, two versions of the questionnaire were distributed, which differ only in their ordering of the scenarios.

In the following diagram the scenarios in the questionnaire (version 1) are represented through the numbers in the streams. For instance, scenario number 6 corresponds with a classic case of negligence; The agent knew about the possible bad outcome and was able to do something about it, but did not make an effort and did not have an excuse. On the other hand, the agent did not intend the bad outcome to occur, and therefore gets attributed less negligence than an agent that willingly lets a bad outcome be realised, such as in scenario 7.



Below is a description of the first scenario.



Scenario 1:
Andre and Bob live in an apartment. Andre has a goldfish, which they see every morning when they eat breakfast. The fish is fed through an automatic feeding mechanism, requiring no human intervention, except for refilling every two months by Andre.

Then the following events occur:
(1) Andre is gone for a week, and Bob is home alone.
(2) Bob notices the fish is acting weird,
(3) but does not think twice about it.
(4) A few days later, the fish is dead.

Questions:
1.1 Please rate Bob's negligence on the following scale,
zero being not negligent, and five being very negligent:

O 0 O 1 O 2 O 3 O 4 O 5

1.2 Which sentence or sentences, ordered by decreasing importance,
contained the reason you used for answering .1?

Sentences …….....

1.3 If you were Bob, describe your feelings towards Andre
by rating the intensity of the following emotions on a scale from 0-5.

Sadness	O 0	O 1	O 2	O 3	O 4	O 5
Happiness	O 0	O 1	O 2	O 3	O 4	O 5
Anger	O 0	O 1	O 2	O 3	O 4	O 5
Guilt	O 0	O 1	O 2	O 3	O 4	O 5
Sympathy	O 0	O 1	O 2	O 3	O 4	O 5
Shame	O 0	O 1	O 2	O 3	O 4	O 5
Fear	O 0	O 1	O 2	O 3	O 4	O 5

1.4 Please do the same for Andre towards Bob.

Sadness	O 0	O 1	O 2	O 3	O 4	O 5
Happiness	O 0	O 1	O 2	O 3	O 4	O 5
Anger	O 0	O 1	O 2	O 3	O 4	O 5
Guilt	O 0	O 1	O 2	O 3	O 4	O 5
Sympathy	O 0	O 1	O 2	O 3	O 4	O 5
Shame	O 0	O 1	O 2	O 3	O 4	O 5
Fear	O 0	O 1	O 2	O 3	O 4	O 5

The next figure gives the varying parts of the other six scenarios.

Scenario 2:
(1) Andre is gone for a week, and
(2) Bob notices the fish is acting weird.
(3) Before he has decided on any course of action concerning the fish,
(4) Bob is called to the bedside of his dying grandmother, forgetting about the fish.
(5) When he comes back, the fish is dead.

Scenario 3:
(1) Andre is gone for a week, and
(2) Bob notices the fish is acting weird.
(3) Bob decides he would have helped the fish,
(4) if only he wasn't committed to playing games on his computer for a week.
(5) After finishing his gaming, the fish is dead.

Scenario 4:
(1) Andre is gone for a week, and
(2) Bob notices the fish is acting weird.
(3) He replaces the fish's water and gives it some fresh food.
(4) After a few days, it is dead.

Scenario 5:
(1) Both Andre and Bob are gone for a week, and
(2) the fish gets sick and dies.

Scenario 6:
(1) Andre is gone for a week, and
(2) the fish gets sick.
(3) Bob notices the fish is very sick,
(4) but doesn't try to help it.
(5) The fish dies.

Scenario 7:
(1) Andre is gone for a week, and the fish gets sick.
(2) Bob notices the fisk is very sick, but
(3) doesn't do anything.
(4) He disliked the fish anyway.
(5) A few days later, the fish is dead.

4.3.2 Predictions

If our model is correct, people will be influenced in the attribution of negligence by the variation of factors present in the different scenarios. To show this, we need to do three things:

1. Show that there is a significant difference in negligence attribution between scenarios.
2. Show that the main reason for this difference is the deviant factor as described in the scenario.
3. Show a correlation between predicted arousal and participant expectation.

For the first item, we need to consider scenario 1. This is, in our model, a very normal case of negligence; Bob noticed the fish was acting weird, and therefore should have known something was wrong. He did not have an excuse, and could have made an effort to save the fish, but didn't. This case will act as our baseline. By comparing the relative negligence ratings between scenario 1 and the other scenarios, we can establish if there is any significant influence of any given factor in human reasoning. Scenario number seven is meant to test one of the extremes of our model; Bob knew the fish was sick, had no excuse and made no effort and actually disliked the fish. In our current model, Bob should receive the highest negligence rating in this scenario.

Because scenario seven differs in two factors (foreknowledge and intention) from scenario one, we cannot directly assess significance by comparing it to this scenario. A comparison between scenarios six and seven allows us to infer the contribution of intention under these circumstances.

In order to minimize workload on participants, we have not made a scenario where there was no foreknowledge but there was intention, and will assume the role played by intention in the case of foreknowledge is similar when there is no definite foreknowledge.

Let the average amount of negligence attributed in each of the seven scenarios be described as $\overline{N}_n$, where n corresponds to the number of the scenario, and the statement $\overline{N}_n < \overline{N}_m$ carries the meaning that average negligence attribution in scenario n is significantly lower than that of scenario m.

Then, in summary the predictions of the first item can be described as follows;

1. $\overline{N}_2 < \overline{N}_1$ In scenario two, Bob is attributed less negligence because he has a good excuse.
2. $\overline{N}_3 < \overline{N}_1$ Bob is attributed less negligence in scenario 3 than in scenario 1, because he has a weak excuse.
3. $\overline{N}_2 < \overline{N}_3$ Agents with good excuses are rated less negligent than weak ones with weak excuses.
4. $\overline{N}_4 < \overline{N}_1$ Because Bob has made an effort, he is given a lower rating than when he has done nothing.
5. $\overline{N}_5 < \overline{N}_1$ In this scenario, Bob could not have known the fish was in a bad condition, and attributed less negligence.
6. $\overline{N}_1 < \overline{N}_6$ Having received explicit foreknowledge of the bad state of the fish in scenario six, Bob is rated as more negligent here than in scenario one.
7. $\overline{N}_1 < \overline{N}_7$ Bob knows about the bad state of the fish and dislikes it. Scenario seven should receive the highest average rating in our questionnaire.
8. $\overline{N}_6 < \overline{N}_7$ We predict that intention of the bad outcome to occur will contribute to negligence rating.

For the second item we will look at the second question in each scenario. This question asks the user to answer which of the given pieces of information in the varying part of each scenario has been most important to the user for

making his attribution of negligence. We predict that in each scenario other than one, the line of information containing the deviant factor compared to scenario one will be identified most frequently as the most important piece of information.

Thirdly, we need to assess consistency between predicted emotional arousal and participant expectation. By looking at the conditions for emotional arousal as presented earlier, we make the following predictions concerning correlations between agent beliefs, attributions of negligence, and generated emotions. Conditions with (neg.) behind them indicate negative correlations.

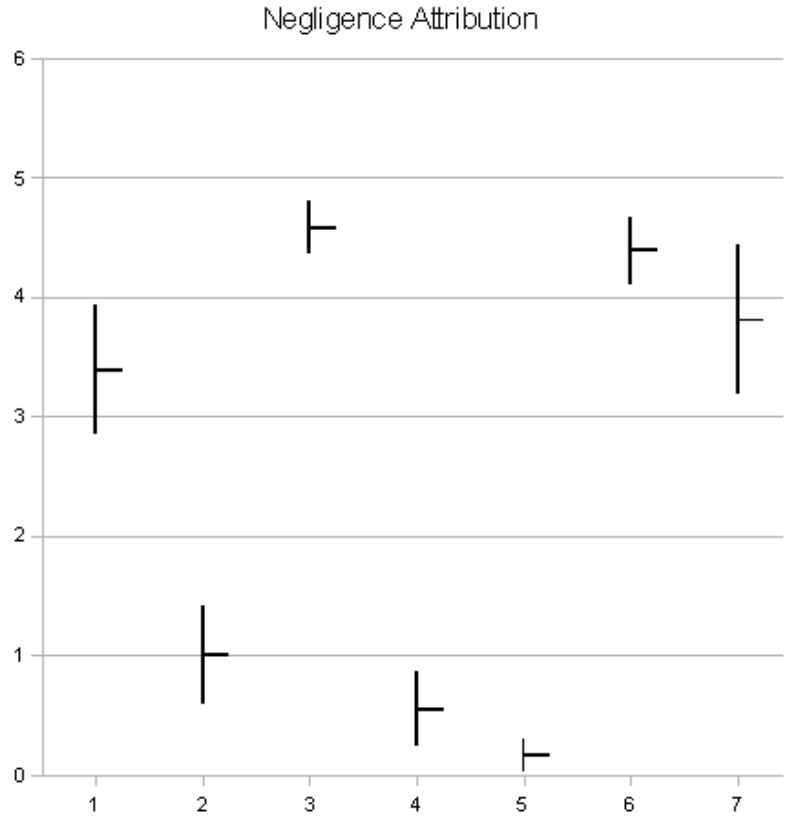
#	Factor	Corresponding emotion
1.	Intention of the outcome	Happiness
2.	Intention of the outcome (neg.)	Sadness
3.	Negligence for another agent	Anger
4.	Negligence for oneself	Guilt
5.	Negligence for oneself	Fear
6.	Another's intention of the outcome (neg.)	Sympathy
7.	Negligence for oneself	Shame
8.	Intention	Shame
9.	Intention (neg.)	Guilt

According to this table, the factors of negligence and intention are primarily responsible for emotional arousal. The correlation for negligence-based emotions will be made based on all scenarios, whereas the correlations for intention-based emotions will be computed based on the results of scenarios six and seven, since they only differ in only one factor, intention.

4.3.3 Results

A total of thirty-one participants responded, of which 19 male (61%) and 12 female (39%). Ages ranged between 21 and 64, with an average of 30.87 and a median age of 28. Participants were mostly staff and graduate students from the ICT (Institute for Creative Technology), ISI (Information Sciences Institute) and USC (University of Southern California). See also Appendix A.

In the diagram below, the average negligence rating for each scenario is given by the horizontal bars, while the vertical bars signify a 95% confidence interval.

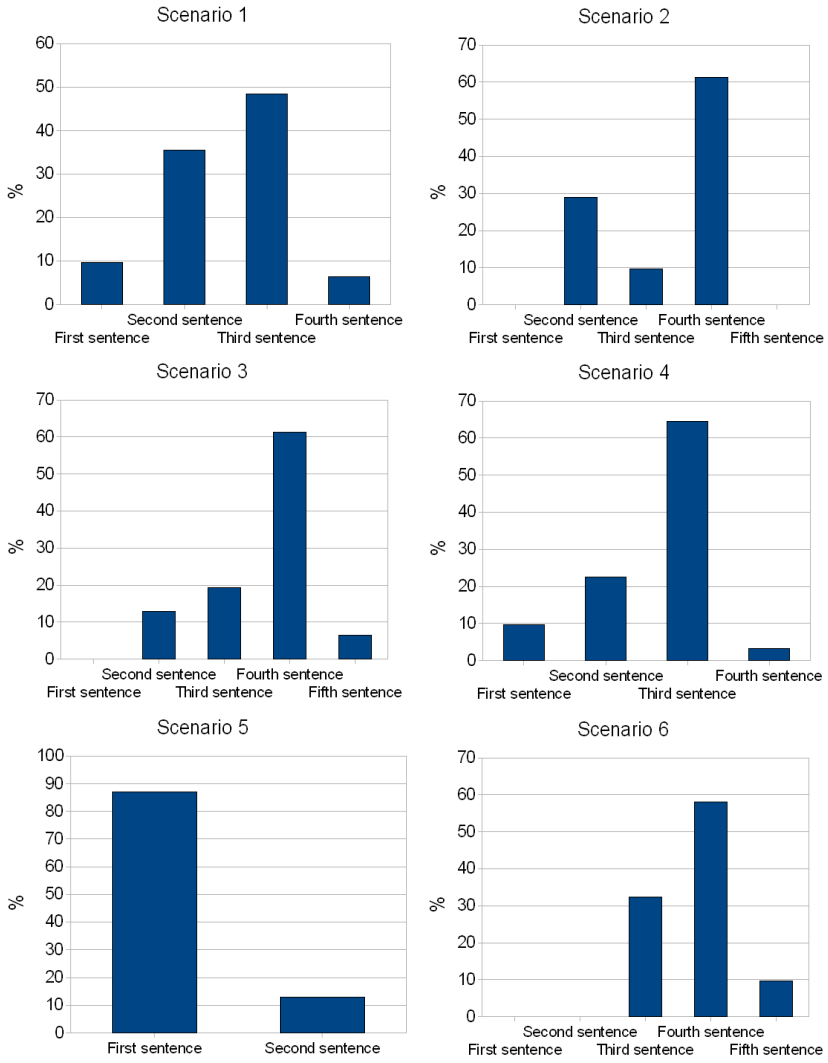


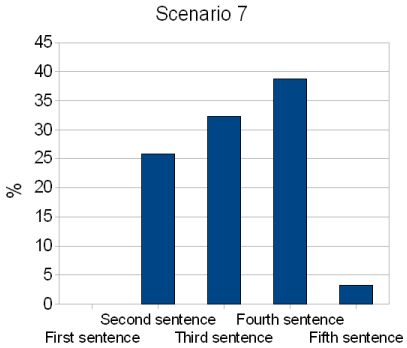
In the previous section, we discussed a number of criteria to hold for our model to be correct. To validate these criteria, we look at the difference in individual negligence ratings between the two scenarios in each equation. For each of these equations, we will test the null-hypotheses, stating that the samples from both scenarios come from an equal distribution. Even though our equations show a prediction in one direction, we are interested in significant deviations in both directions, and will therefore apply a two-tailed t-test. Application of this test yields p-values describing the likelihood of the null-hypothesis being true. Low p-values allow us to reject the null-hypothesis and conclude a significant difference between populations exists. Results are given below (values are rounded to six figures):

Results		
#	*relation*	p
1	$\overline{N}_2 < \overline{N}_1$	0,000000
2	$\overline{N}_3 < \overline{N}_1$	0,000060
3	$\overline{N}_2 < \overline{N}_3$	0,000000
4	$\overline{N}_4 < \overline{N}_1$	0,000000
5	$\overline{N}_5 < \overline{N}_1$	0,000000
6	$\overline{N}_1 < \overline{N}_6$	0,000252
7	$\overline{N}_1 < \overline{N}_7$	0,222844
8	$\overline{N}_6 < \overline{N}_7$	0,074088

Using a 5% significance level, we have to reject most null-hypotheses, with the exception of equation seven and eight. Also, our predictions in equation two and eight are reversed. In equation two, we estimated that someone with a bad excuse has less negligence than someone without any excuse. Finding a significant difference in the other direction means our model has to take into account that people will increase negligence ratings for a bad excuse, but decrease it for a good excuse. Also, people judged a person with foreknowledge and intention of a bad outcome as slightly less negligent than a person with only foreknowledge.

Next we will look at the information people have used in their attributions. In the second question of each scenario, people are asked to suppply which sentence(s) they thought was most important for the attributions they made concerning negligence. The following diagrams show distributions for each scenario of what people thought was the most important sentence each scenario. Using the questionnaire descriptions in 4.3.1, we can determine which sentences belong to which bars.



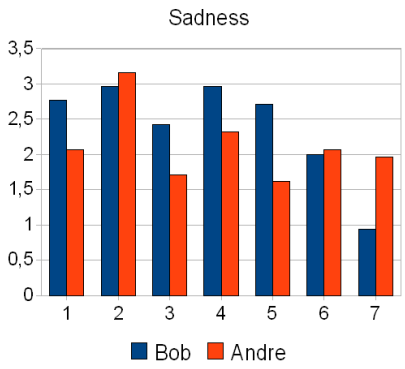


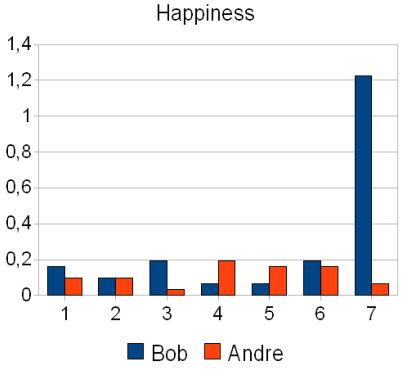
As we can see, in scenario one, our base example, people value sentence number three the most valuable, in which Bob is described as doing nothing about the fish acting weird, followed by sentence two, where he notices the fish is acting weird. These sentences correspond in our model to not making any effort and having foreseeable knowledge. Scenario two is even more clear; Sentence four is chosen by almost two-thirds of respondents. In this sentence, Bob is called to the bed of his dying grandmother, corresponding to having a strong excuse in our model. A similar effect can be seen in scenario three, where people respond most strongly to Bob having the excuse of wanting to play video games for a week. In scenario four, where we see a significantly smaller attribution of negligence, people have chosen sentence three as being most important. In this sentence, Bob makes an effort by replacing the fish's water and giving it some fresh food. People chose sentence two as being most important in scenario five, where Bob is described as being away from the fish for a period of time. This corresponds in our model to lack of ability. In scenario six, people choose sentences four and three; Bob doesn't do anything about the sick fish but has definite foreknowledge. Scenario seven adds the component of intention. The ratings are more evenly distributed between Bob disliking the fish, and having knowledge about the fish being sick and not doing anything about it like in scenario six.

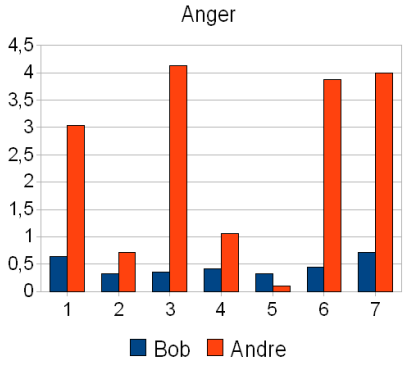
Overall, as very few people rated the outcome of the scenario as the most important piece of information, we can say that people are not so much interested in the results of the negligence, but more in the actions of the negligent party and the circumstances under which they occurred. Furthermore, we see that in every scenario where Bob's foreknowledge and resulting actions are described, both are perceived as important but participants more frequently chose the resulting action as the most important piece of information. In scenarios where excuses are present, the sentences describing the excuses are rated as most important, strengthening our belief of the influence of this factor. All in all, we can see that in most scenarios, the factor present in the sentence judged by participants to be the most important corresponds with an expected change in negligence rating for that factor, when compared with the base scenario, suggesting the correct estimation of this factor. On the contrary, in scenario seven, where Bob dislikes the fish, this factor is judged by participants as being the most important sentence, but this is hardly reflected in the negligence rating.

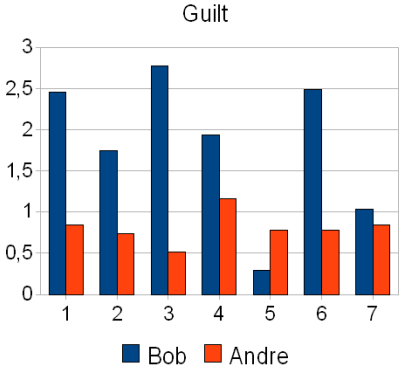
Next, we shall look at the expected emotional arousal as reported by participants. Below are presented the average reported emotional states of participants for both characters. These are followed by a diagram of the 95% confidence in-

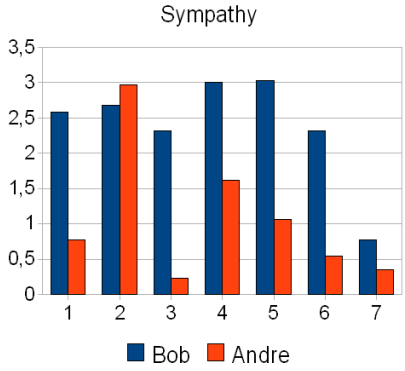
tervals of predicted correlations between selected factors and emotional arousal.

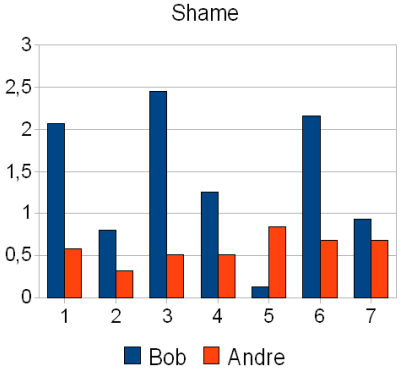


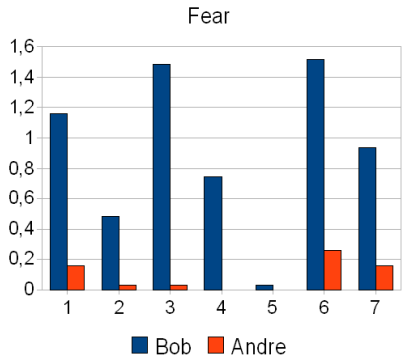


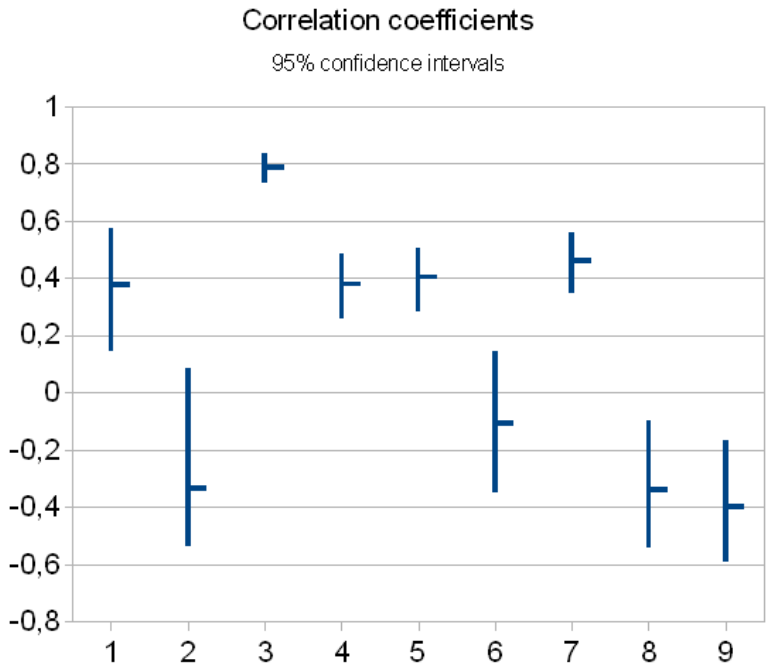












Correlation coefficients corresponding to predicted emotional arousal as described in 4.3.2

A cursory glance on the diagram above tells us that most of the predicted correlations were correct. Even though an agent might know a specific outcome to be appreciated negatively by a fellow agent, if it was intended by the agent himself, the outcome still brings considerable happiness. As shown in [Feigenson 2001], perceived negligence from another agent arouses anger towards that agent. If the negligence is caused by oneself, the agent feels guilt, fear and shame. Intention of an outcome that is negative for another agent reduces guilt, but contrary to predicted also reduces shame. A possible explanation for this is that participants valued an implicitly lazy agent such as in scenario six as more shameful than an agent with explicit bad intentions such as in scenario seven. From these results, it is not clear whether sadness and sympathy are negatively influenced by the intentions of agents.

4.4 Conclusions

Having taken inspiration from psychological, legal and economical theories as well as common sense reasoning, we have developed a model that predicts how certain factors in social environments influence human atributions of negligence. After testing, we find that our model is able to predict most of these influences correctly; Agents that had the ability or knowledge to prevent a bad outcome are rated more negligent, just like agents that made no effort to prevent it versus an agent that did.

More interesting is the case of having an excuse: According to the results in the previous paragraph, humans respond both to weak as well as strong excuses, but in different directions. While a strong or good excuse lowers negligence ratings as expected, weak ones will increase it. We currently have no strong indication why this is, but it is possible people perceive an agent with a bad excuse as selfish or dishonest and project these unsocial character traits onto the negligence rating.

5 Computing negligence

Using the attributional model of negligence as developed in the previous chapter, this chapter presents a model of negligence determination as implementable in an agent system. Before we can define how an agent is to know and reason about negligence concerning itself or other agents, we have to define the environment in which the agent will function. Most of the data structures present in our definition of this environment take on the form of relationships between various entities. This makes our model well-suited to be expressed in terms like those of tuple relational calculus [Codd 1970].

To illustrate this notation, we shall now provide a few examples. Let's say that we define an employee in our system consisting of a name, an address, and a function. We then represent the structure of the *employee* record as a 3-tuple:

$$employee = (name, address, function)$$

The name of an individual employee e can be referenced using dot-notation:

$$e.name$$

When we then have a set of employees Emp, and we want to address the subset of employees R that has a research position, we can describe this using traditional set notation;

$$R = \{e \mid e \in Emp \wedge e.function = researcher\}$$

In the rest of this chapter, there will be a collection of examples to illustrate the workings of our model. These examples provide the reasoning done *by* an agent *about* another agent. By convention, let the agent that is doing the reasoning be addressed as i, and the agent *about* whom the reasoning is being done be referred to as g. It is possible but not becessary that i and g are the same agent. In this case, i is thinking about his own actions.

In 5.1, we will start by describing our agents' top-level data structure, the world view. We will then continue by describing its components in more detail, drilling down until we have a full understanding of an agent's view of the world.

Section 5.2 will contain a bit more detail regarding the conclusions an agent is able to draw from certain world states, and finally in 5.3 we will describe the algorithm agents use to form opinions of negligence concerning one another.

5.1 The Agent Environment

The agent has a view of its environment which we shall refer to as world view. This view is the interpretation of the world by the agent. Indeed, world view needs not to be a complete or even correct representation of the world on which the agent exerts its influence. We shall refer to the agent's world view as Ag. World view consists of a 3-tuple of a number of ordered World states W, past and present, an action library *actions*, and a set of *conditions*.

$$\text{Ag} = (\text{ World states, actions, conditions })$$

A world state $w_n \in W$ is the representation the agent had of the world at a single moment in time. One such state can be broken down into a 6-tuple of the agent identity i, a set of agents G, a set of variables $variables$, a reference to a point in time $time$, a set of mutations in relation to the previous world state $mutations$ and a set of partially completed actions $partials$ at that point in time.

$$w_n = (i, G, variables, time, mutations, partials)$$

These will be discussed in further detail in the next sections.

5.1.1 Variables

A variable $v \in w_n.variables$ describes an aspect of world state that is relevant to an agent, and is a named entity (a 2-tuple) that can take on a value, or have an *undefined* value. Possible values for a variable needn't be known to the agent on forehand. Because a variable is central to a specific agent, it is possible for multiple agents to have sets of variables that are completely or partially overlapping, or disjoint.

Example of a set of variables V:

Variables	
Name	Value
Location	home
Front Door	closed
Lamp	*Undefined*
Kittens	4
Current Time	14:01
Bedroom light	off
Funds	$200,-
Goldfish health	Good
Fish water state	Clean

World state is characterized by partial observability of these variables. Analogous to real life, an agent will not always be able to evaluate the value of a variable right away: We are not omniscient, and cannot know what happens in a place when we are not there. This is reflected in our model by associating an aspect of visibility to variables, which influences the interpretation of a variable by the agent. Given this aspect, it is possible for a variable to be in one of three states:

1. Void: The variable has never been observed. It does not have to be clear whether the variable can be observed.
2. Visible: The variable is visible to the agent, allowing the agent to evaluate the current value of it.
3. Invisible: The variable used to be visible, but isn't anymore. The agent cannot inspect the current value.

In order to be able to describe how visibility of a variable is determined in our model, we need to describe conditions, which we will discuss next.

5.1.2 Conditions

A condition or variable-condition $c \in conditions$ is a named triplet of a variable name *name*, a relationship *relation*, and a value *value*. If the value of the variable satisfies the relationship to the value (*variable relation value* evaluates to true), then the variable-condition is defined as satisfied.

Example of a set of variable-conditions:

Variables-conditions			
Name	Variable	Relationship	Value
Have money	Funds	greater than	$\$150, -$
At home	Location	equals	home
Bedroom light on	Bedroom light	equals	on
Alice gone	Location Alice	unequals	Home
Box lid open	Box lid	equals	open
Water clean	Fish water state	equals	clean
Water not clean	Fish water state	unequals	clean
Water not dirty	Fish water state	unequals	dirty
Water not very dirty	Fish water state	unequals	very dirty
Water very dirty	Fish water state	equals	very dirty
Fish healthy	Goldfish health	equals	Good
Fish sick	Goldfish health	equals	Sick
Fish dead	Goldfish health	equals	Dead

Let us define a function *evaluate* which tests whether a condition is satisfied:

$$evaluate : Worldstate \times condition \mapsto boolean$$

Using the example conditions above, we can see that if the variable goldfish health is equal to good, $evaluate(fish\ healthy)$ evaluates to $true$.

5.1.3 Visibility

Variables that are void or invisible have, by definition, an *undefined* value. Visibility of a variable for an agent is determined by zero or more visibility-conditions, describing restrictions on world state to hold for the variable to be visible to the agent. As such, visibility-conditions are part of the agent description $g \in G$, which we will see later on in this chapter.

A visibility-condition can be described by a tuple of the variable name *name* and a set of variable-conditions *conditions*. If an agent g has no visibility-conditions for a specific variable then that variable is always visible for g. Below is the protytype for a visibility-condition as well as a table of example conditions.

$$visbilitycondition = (name, conditions)$$

Example of a set of visibility-conditions VC for $g \in G$:

Visibility-conditions	
Variable	Variable-conditions
Box lid	{At home, Bedroom light on}
Box contents	{Box lid open}
Front Door	{In front of house}
Front Door	{At home}
Goldfish	{At home}
Fish water state	{Water very dirty}

A visibility-condition is defined as satisfied when all of its variable-conditions are. We can generalize the function *evaluate* from the previous section to apply to visibility-conditions:

$$evaluate : Worldstate \times visibilitycondition \mapsto boolean$$

evaluate returns true for any visibility-condition vc in world state w_n when there are no unsatisfied preconditions:

$$evaluate(w_n, vc) = true \Leftrightarrow \neg\exists c \in vc.conditions : evaluate(c) = false$$

In the previous example table, the agent can only observe the status of the box lid when it is at home and the light is switched on. On the other hand, a variable can also have multiple visibility-conditions. In the same example, 'Front Door' has two of them. Contrary to visibility-conditions, which need all of their variable-conditions to evaluate to true to be satisfied, a variable needs only one visibility-condition to evaluate to true to be visible. Thus, in the previous example, g is able to inspect the state of the front door when he is at home, or when he is standing in front of his house. This nested approach allows us to specify the visibility of a variable in a powerful way using logical disjunctive normal form (DNF), and lets us use the function *evaluate* to define the visibility of a variable as a function *visible*:

$$visible : Worldstate \times agent \times variable \mapsto boolean$$

The definition of this function is given below:

$$visible(w_n, g, v) = true \Leftrightarrow \exists vc \in g.visibilityconditions : evaluate(w_n, vc) = true \wedge vc.variable = v$$

This is not all. In the previous section, we have defined visibility-conditions as defining visibility of variables. However, visibility-conditions can themselves be dependent upon invisible variables, in which case we cannot know whether we can know the value of a variable.

An example of this can also be found in the previous section: In order to inspect the box contents, the box lid must be open and the agent must be near the box. But, when the agent is not at home or the light is not switched on, the agent has no visibility over some of the variables needed to inspect the value of the box lid. What is the current value of the box contents? The prudent solution is to say that we cannot say anything about the box contents and define it as *undefined*.

In general, we state that any visibility-condition that relies on an invisible variable in one of its variable-conditions is defined as *undefined* or *invisible*. By recursion, any variables that have only invisible visibility-conditions are consequently also defined as *invisible*.

When confronted with a problem that requires reasoning about invisible variables, an agent can apply several strategies. The agent can simply maintain the last known value, develop a statistical approach for a likely value, or infer the value from other, visible variables in world state. Such functions fall out side of the scope of this research.

5.1.4 Motivation

Agents are assumed to be goal-oriented creatures, wishing to set variables from world state to a specific set of values. States that are desirable can be distinguished from undesirable ones by the *utility* or their variables. A higher utilityrating means a state is more desirable than a state with a lower utilityrating. Variables that are absent from the utility function are assumed to be indifferent to the agent and receive a value of zero regardless of their value. An example of a utility function for g is given below:

Example utility function for g:

Utility(g)		
Name	Value	Utility
Health	Good	1000
Health	Bad	700
Health	Dead	0
Fish	Healthy	20
Fish	Weird	18
Fish	Sick	15
Fish	Dead	0

To calculate the relative utility of a mutation, we shall introduce a new function, named *utility*. This function takes as arguments an agent, a variable name, and a value, and returns us the utilityrating this particular agent has attached to that variable having that value. Its prototype is given below:

$$utility : agent \times variable \times value \mapsto utilityrating$$

For example, when agent g is in good health (utility: 1000) and then gets sick (utility: 700), the agent experiences a relative utility of utility(g, Health, Good) - utility(g, Health, Bad) = -300.

5.1.5 Actions

The agent possesses a database of knowledge about named possible actions *actions*. The information contained in in an action tells the agent about which agent g is able to execute each named action *name*, what the preconditions *conditions* on world state are and the actions' effects can be, the *cost*, *duration*, execution-visibility *visible* and the execution-likelihood p when its preconditions are met. This can be represented by an 8-tuple:

$$action = (name, agent, conditions, effects, cost, duration, visible, p)$$

The preconditions for a particular action are simply variable-conditions as defined in the previous section. If there are no unsatisfied preconditions, the action can be executed and can be referred to as *possible*. Let us also define a new function with the same name:

$$possible : Worldstate \times action \mapsto boolean$$

possible returns true when there are no unsatisfied preconditions:

$$possible(w_n, a) = true \Leftrightarrow \forall c \in a.conditions : evaluate(w_n, c) = true$$

5.1.6 Invisible actions and nature

Since we know that the contamination of the water is the result of a bacteriological process, it would be elaborate to include an agent in our model that represented the water-borne bacteria that cause the contamination. Instead, this action is described as being executed by an impersonal agent *nature*. In our design, the agent *nature* is a special, impersonal agent that is responsible for any actions that are normally not associated with an individual. These actions may include any actions brought about by the laws of nature or bacteriological or chemical processes. For example: actions that are weather-related; lightning, rain, floods, storms as well as fire hazards are typically actions that would belong to *nature*.

Invisible actions are typically executable only by *nature* and cannot be directly observed. They can however afterwards be deduced from the occurrence of their effects, or be expected beforehand when there is a high execution-likelihood in combination with satisfied preconditions for a specific action.

The relevance of invisible actions comes forward when we revisit the actions database. As we can see, the water for the fish can be clean or dirty. Dirty water satisfies a precondition that allows the fish to become sick more easily. While the agent cannot observe the water getting dirty (unless it gets really dirty), the agent must still account for the likelihood of cleaning dirty water to keep the fish healthy.

But how can we form an expectation about when such an action will be executed? For an action that is associated with a normal agent, we may know the intentions of the executing agent, or are able to identify them based on previous observations. This gives us a way to estimate whether a certain action is likely to be executed in the future. For actions executed by *nature*, we use a different mechanism; *likelihood*. As can be seen in the actions database, every action has a likelihood entered in the last column. This represents the chance that a certain action will be executed by *nature* per unit of time if it's preconditions are fulfilled, and allows us to form an estimate of certain risks and dangers involved.

Example:

For this example we will look at the actions database on the next page. We pay special attention to two actions; a_1: 'Fish becomes sick 1', and a_2:

Example of an action database A:

Actions							
Name	Agent	Precond.	Effects	cost	duration	visible	likelihood
Go to city from home	*g*	{At home}	{Arrive at city}	20	0.02	true	1.0
Go home from city	*g*	{In city}	{Arrive home}	20	0.02	true	1.0
Water becomes clean	*g*	{Water not clean}	{Water clean}	0	0.03	false	1.0
Water becomes dirty	*nature*	{Water not dirty}	{Water dirty}	0	2	false	0.3
Water becomes very dirty	*nature*	{Water not very dirty}	{Water very dirty}	0	4	false	0.3
Become sick	*nature*	{Healthy, Disease present}	{Get sick}	0	2	false	0.3
Healthy fish becomes sick 1	*nature*	{Fish healthy, Water clean}	{Fish gets sick}	0	2	false	0.02
Healthy fish becomes sick 2	*nature*	{Fish healthy, Water not clean}	{Fish gets sick}	0	2	false	0.15
Sick fish gets better	*nature*	{Fish sick, Water clean}	{Fish gets healthy}	0	2	false	0.70
Healthy fish dies	*nature*	{Fish healthy, Water not clean}	{Fish gets sick}	0	1	false	0.05
Sick fish dies	*nature*	{Fish sick, Water not clean}	{Fish gets sick}	0	1	false	0.70

'Fish becomes sick 2'. These actions roughly represent the same thing; The fish becomes sick, but they happen under different circumstances (this can be seen by their preconditions): In #1, the fish has clean water. In #2 the fish has water that is not clean.

Let's say that agent g is going to go away for some time, and is going to leave the fish alone for a while. This means the fish is not a visible variable for g while away form home. If g wants to estimate the chance that the fish has started to become sick after he returns, he can use these likelihoods to make his calculation: If g assumes the water will stay clean and is away for n units of time, using simple probability mathematics, the chance that the fish has started to become sick when he returns will be equal to

$$1 - (1 - a_1.likelihood)^n.$$

For a visit of 3 units of time, this would translate into a chance of

$$1 - (1 - 0.02)^3 = 0.06.$$

However, if g were to assume that the water wasn't clean when he left, this chance would be considerably higher:

$$1 - (1 - 0.15)^3 = 0.39.$$

From this example, it is immediately clear that if g wants his fish to remain healthy, that he should clean the fish water before he leaves on a long trip.

For actions that can be executed by other agents than *nature*, the likelihood can be interpreted as the chance that the agent is successful in executing the action when it attempts to do so. It still does not state, however, that the succesfully executed action will have an effect that will result in an outcome that changes world state.

5.1.7 Effects

Actions change world state by causing changes in variables. We can describe the changing of a variable to a certain value as an effect. The moment at which the variable changes we can say the effect is happening or occurring.

Every action has a non-empty set of effects *effects* associated with it. The only time at which these effects *may* occur is at the moment when the action is completed. The likelihood of an individual effect e occurring at this time is encoded in the effect by a probability p. If e does occur, it generates a change in world state w_n called a mutation or outcome, thereby also triggering the creation of a new world state, w_{n+1}. This new world state will then have a different value for the variable described in the effect. Below is our definition for an effect and an example effect set:

$$effect = (name, variable, value, p)$$

Example of an effect set:

Effects			
Name	Variable	Value	p
Arrive home	Location	Home	1.0
Arrive at city	Location	City	1.0
Get sick	Health	Bad	0.5
Fish gets sick	Fish	Sick	0.5
Fish gets better	Fish	Healthy	0.5
Fish dies	Fish	Dead	0.5

Note that a typical effect set for any action will probably not contain this many entries, and just contain maybe one or two rows. The multitude present in the previous table is mainly for illustration purposes. An effect may be present in more than one action. To find the set of actions that can cause a certain effect e, we shall create a new function:

$$causesof : effect \mapsto \{Actions\}$$

This function is defined as follows:

$$causesof(e) = \{a \mid e \in a.effects, \text{ where } a \in W.actions\}$$

In other words, *causesof* returns the set of actions that can generate e. When examining a certain world state, it might be useful to know whether it's possible to execute an action which generates effect e. For this purpose, we can overload the function *possible* to also accept an effect as a second argument:

$$possible : Worldstate \times effect \mapsto boolean$$

with the associated function definition:

$$possible(w_n, e) = true \Leftrightarrow \exists a \in causesof(e) \wedge possible(w_n, a) = true$$

When we combine an effect with a world state, we can calculate the expected utility for an agent g:

$$utility : Worldstate \times agent \times effect \mapsto utilityrating$$

$$utility(w, g, e) = (utility(g, v, e.value) - utility(g, v, v.value)) * e.p, where\ v \in w.variables \wedge v = e.variable$$

In the next section, we will see how the occurrence of effects leads to the creation of new world states.

5.1.8 Mutation of world state

As explained in the beginning of this chapter, W contains the set of world states, describing the agent view of the world at a certain instant in time. Initially, W contains only one element; w_0. Initiation or completion of an action a in this state triggers a change in world state, leading to the creation of a new state w_1. In general, we can say that initiation or completion of an action in state w_n leads to the creation of a new state w_{n+1}, which is added to W.

An example of this would be that if at state w_0 at time $t_0 + interval$ agent g initiated the action *Go home from city*, g would trigger the creation of a new state w_1 in which the action *Go home from city* is partially completed. This action is then present in the list of partial actions $w_1.partials$ with the time at which it was initiated, $t_1 = t_0 + interval$. For the rest, w_1 is simply a copy of w_0.

Example of the partial action set $w_1.partials$:

Partial actions:	
Action	Time
Go home from city	t_1

After a set amount of time (in this case, 0.02 units of time according to the 'Go home from city'-entry in the action database), the action *Go home from city* completes and this triggers the creation of another world state, w_2. We model the creation of a new world state as the result of a set of mutations as a function:

$$application : Worldstate \times \{Mutation\} \mapsto Worldstate$$

This function changes the value of the variables denoted by the set of mutations to their new values.

The new state w_2 is a copy of w_1, except that because the action has completed, the action does not appear in the new partial actions list, $w_2.partials$. Instead, the action has generated a mutation: The location of g, represented by the variable *Location*, has changed state from 'city' to 'home'. In this case, we are certain that executing the action 'Go home from city' will achieve the result of changing the location from 'city' to 'home', as we can see by the probability of 1.0 for this effect (traffic accidents and such have not yet been accounted for in this model).

However, for some actions, the probability p will not be 1.0, and the world state after the action is completed is not *a priori* clear. For these kind of actions, an agent wanting to predict the outcome of an action will have to maintain a probability distribution over a set of states instead of relying on a single state. We will get to know more on this subject in 5.2.

The changing of location from 'home' to 'city' is represented in the set of mutations $w_2.mutations$, which is described below;

Example of the mutation set $w_2.mutations$:

Mutations for state w_2			
Name	Variable	Old value	New value
Arrive home	Location	city	home

Note that the name of the mutation is always equal to that of the effect that triggered the mutation. At this point the reader might wonder as to what the difference is between an effect and a mutation. The answer to this question is

that an effect describes a *possible* change of state with an estimated probability with which this might happen, while a mutation describes a definite change of state as the result of an effect triggering this change.

Nevertheless, even though an effect may be present in multiple different actions, an effect can only generate one specific mutation and a mutation can only be generated by one specific effect. When a mutation m is the result of an effect e we shall say the following relation evaluates to true:

$$m = e$$

In this section, we have described how g is able to execute the action 'Go home from city' and change it's location from 'city' to 'home'. To summarize the execution of 'Go home from city' we can provide a trace that shows how the action has developed through these three world states into an mutation and has changed the value of the variable 'Location':

Trace of the action 'Go home from city'			
State	Partial actions	Mutations	Location
w_0	$\emptyset$	$\emptyset$	city
w_1	{Go home from city}	$\emptyset$	city
w_2	$\emptyset$	Arrive home	home

We are now ready to expand our utility function defined earlier. Instead of having to write $utility(g, variable, newvalue) - utility(g, variable, oldvalue)$ every time we want to know the relative utility of a mutation, we shall define the utility of a mutation $m \in w_n.mutations$ for agent g as follows:

$utility : Agent \times Mutation \mapsto utilityrating$

$utility(g, m) =$

$utility(g, m.variable, m.newvalue) - utility(g, m.variable, m.oldvalue)$

5.1.9 Opinions

In the previous sections we have explained how our agents view the world, the actions that happen in it and how these actions affect their perceived environment. Combining this with a per-agent utility function gives our model the ingredients it needs to let agents evolve a step further and let them develop their own opinions about what's going on in their world.

Because we concentrate on the subject of negligence, an opinion of an agent g will always be associated with both a target agent $h \in G$ and a mutation $m \in w_n.mutations$. Such an opinion reveals how g feels about h because of m happening and consists of a negligence rating and a set of emotions. In some cases, g and h may be the same agent. Because we shall make an opinion a part of a target agent definition, we do not need to specify the target agent in the definition of the opinion itself. We still need to include the reference to the mutation, however. Below is our prototype for an opinion:

$$opinion = (mutation, negligence, emotions)$$

Of course the mutation can only have occurred as the result of an action. Because we are mainly interested in negligence here, the referenced mutation need not be (and usually isn't) caused by either agent for an opinion to be formed by one agent about another.

The set of emotions present in the current model is, not by chance, the same set of emotions we have studied in chapter four. The negligence rating is encoded as a number between zero and five inclusive (like in the questionnaire). The interpretation of a rating can be in one of four categories: *Minimal*, *Low*, *Medium* or *High*. *Minimal* is reserved for ratings of 0.5 and lower, *Low* is for ratings between 0.5 and 2.0, *Medium* is for everything in between 2.0 and 4.0, and everything above 4.0 is considered *High*. Individual emotions are encoded in the same way as variables; They are a 2-tuple of a name and a value. The possible values for individual emotions are also a numeric value between zero and five. Not all emotions need to be present in a single opinion.

Emotion prototype:

$$emotion = (emotion - type, value)$$

Like in most of this chapter, we accompany model descriptions with examples. In this example, we will concentrate on aspects that are related to opinions and will leave out most of the other specifications of the enviroment.

Example:

We find agent g coming home after a weekend away to find his apartment partially burned down (mutation 'apartment burned'). g, who lives together with h, had asked h to let the faulty electric wiring be fixed before he would return. Even though h did not set fire to the house (*nature* did), g develops an opinion about h:

mutation	negligence rating	emotions
apartment burned	5 (High)	{(anger, 5), (sadness, 4), (sympathy, 1), (fear, 3)}

Now that we have defined how an opinion is related to agents and what is happening around them, we are ready to put these pieces together and present a complete definition for an agent's idea of another agent in its world. This is the subject of the next section.

5.1.10 Other Agents

Naturally, one of the most important data structures is an agents' representation of other agents. When we speak about an agent representation, we are talking about the image an agent has of another agent, not the entire world view of a single agent. In our system, we acknowledge that different agents can have different utility functions, so these are included in the agent representation. Furthermore, variables which are visible to some agents in state w_n, will not be visible to other agents (knowledge is not universal). This is encoded in the fact that *visibility conditions* are also agent-dependent and thus part of an agent description. We also model agents as being resource-bounded in the sense that

they, like humans, will not have unlimited computational resources. This is represented by a deliberation time *delib*, signifying the time an agent can spend processing new information upon world state changes before it is expected to react. This does not mean an agent will necessarily need this time to react to an event, but rather an upper bound after which we can expect it to have deliberated sufficiently to have formed a new plan. Lastly, an agent definition also consists of a set of opinions about that agent as defined in 5.1.10.

After taking the above into account, we have come to the following definition of an agent representation:

$$g = (name, utility, visibility\ conditions, delib, opinions)$$

Our next example is a rather special one, because it is the first time we define a complete agent representation. Our agent's name is Harry. His representation is a part of Sue's world view, so this is the way Sue thinks about Harry. Due to the complexity of defining a complete agent representation, we will keep Harry a rather simple agent.

$$Harry = (Harry, utility_h, viscond_h, delib_h, opinions_h)$$

where $utility_h$ is defined as:

$utility_h$		
Name	Value	Utility
Activity	working	0
Activity	sleeping	10
Activity	watching television	20
Location	home	15
Work-status	employed	150
Work-status	unemployed	-150
Location	home	20

From this we can see that Harry cares for only a few things in life: He likes sleeping but ideally he wants to be watching television at home all day. He really hates being unemployed, though, so we can expect that even though Harry doesn't like to work, sometimes we can find him working.

We've kept the visibility-conditions rather simple but sufficient for illustration purposes:

$viscond_h$	
Variable	Variable-conditions
Harry's television	{Harry at home}

Harry's deliberation time $delib_h$ is set to 15 minutes and the opinions Sue has about Harry are captured in $opinions_h$:

$opinions_h$		
mutation	negligence rating	emotions
Car broke down	3	{(anger, 3), (sadness, 3)}

From this we can see that Sue thinks Harry is fairly negligent because of a mutation about a car that broke down. She seems to be quite angry at him over this, but also feels quite sad, because she might have expected Harry to have fixed her car but sadly, he let her down again.

At this point, however, we can only speculate as to the circumstances under which this opinion has arisen, because this is just an agent representation. In this example, we don't have any information about the world states in which this opinion has been formed.

Finally, it is also important to realize that an agent usually also has an agent representation about itself, and that this agent representation can also include opinions about the agent itself. For example, we might consider that agent Harry has almost the same opinion about himself about the car breaking down; He might consider himself negligent, and be both angry and sad about it. However, because this is a case of an agent evaluating itself as feeling negligent, we would also expect Harry to feel guilty and or ashamed of himself, creating the following opinion:

$opinion_{hh}$		
mutation	negligence rating	emotions
Car broke down	3	{(anger, 1), (sadness, 2) (guilt, 4), (shame, 2)}

Note that this last opinion, $opinion_{hh}$, is an opinion which Harry has about himself, and as such does not exist in Sue's world view.

5.1.11 Discussion

In this section, we have explained how our agents model the world. We have received an in-depth look at their datastructures and understand the way they are able to model change in their environment. In the next section, we will use this model to present some basic reasoning skills our agents will use to assess a situation, and predict the occurrence of future events.

5.2 Basic reasoning skills

In the previous section, we have concentrated on defining the data structures needed for our agent to represent its world in sufficient detail for it to be able to reason about negligence. In this section, we will define a few reasoning steps or functions an agent can use to obtain a basic understanding of a certain situation. We will introduce two new important concepts; *blocking* and the *negligent interval*. These concepts will play a central role in 5.3, where we will use it as a base to start making conclusions about possible negligence.

5.2.1 Blocking

In our model, when an action a is executed, this will likely lead to an effect that will change world state. If world state can be changed by a in such a way that successful execution of another action, b, is less likely to succeed than before a was executed, we can say that a blocks b. In this section we will explain what we mean by the phenomenon *blocking*, how we can find out when a certain action blocks another, and why understanding blocking is important.

At its most basic level, blocking works by letting an effect invalidate a condition. We incorporate this feature into our model by creating a new function:

$$invalidates : Worldstate \times effect \times condition \mapsto boolean$$

with a straightforward definition:

$invalidates(w, e, c) = true \Leftrightarrow$ evaluate(w,c) = true $\wedge$ e.variable-name = c.variable-name $\wedge$ evaluate(c.variable-name c.relation e.value) = false

invalidates returns true for a combination of a condition c and an effect e in world state w_n when both c and e reference the same variable, and the value encoded in *e.value* invalidates the relation encoded in c. Note that *invalidates* returns false when the condition is already unsatisfied: Only effects that change a condition from true to false will receive a value of true from the *invalidates* function. We can use *invalidates* to create another function that checks whether one action blocks another:

$$blocks : Worldstate \times Actions \times Action \mapsto boolean$$

again with a straightforward definition:

$$blocks(w, a, b) = true \Leftrightarrow \exists e \in a.effects \wedge \exists c \in b.conditions : invalidates(w, e, c)$$

A more difficult task, though is the evaluation of whether an action blocks an outcome o. While it certainly would be easy to say that an action a blocks another action which could cause o and therefore a blocks o, this would not always be true. An outcome can be caused by more than one action. While the changing of a variable might block one action that can cause o, another might be unblocked by this change in state. When the unblocked action has a higher execution chance than the action that was blocked, it is actually more likely that o will follow than before.

Clearly, a better approach is needed. What we need is a function that allows us to view the change in likelihood of o occurring as the result of an action. To

accomplish this, we will first define a few helper functions. Because we are only interested in negligence and in our model the relevant actions for this are usually executed by *nature*, this allows us to simplify our calculation somewhat by overlooking actions by other agents. Note that it is still possible to make this calculation for actions executed by any agent once a model of the likelihood of these actions is developed. However, developing such a model for other agents is often agent-specific and falls outside of the scope of this thesis. First, we need a function that evaluates the chance that o will occur as a result of an action being executed in the next unit of time.

$$outcome\_probability : Worldstate \times mutation \mapsto probability$$

We can use the probabilities p encoded in the actions database to estimate what *nature* will do. Using theory from probability calculus, we define *outcome_probability* as:

$$outcome\_probability(w_n, o) = \\ 1 - \prod\{1 - (a.p * e.p) \mid a \in causesof(o) \ \wedge a.agent = nature \wedge \ possible(w_n, a) = true\}$$

In this function, $causesof$ gives us the set of actions that can cause o. Using the product of the negated probability $a.p$ from every action in this set multiplied by the realization chance of the effect $e.p$ we are able to estimate the chance that none of these actions will cause e. Negation gives us the chance that at least one of these actions will produce a realization of e.

outcome_probability has given us the tools to compare the likelihoods of o in different situations, but we are not there yet: execution of an action does a not consistently produce the same end result in the same situation each time. Instead, every effect $e \in a.effects$ has a realization chance $e.p$, describing the chance with which a mutation may or may not occur.

An *a priori* simulation of the execution of an action will therefore not reveal a single state, but rather a set of states. Each of these states might have a different set of possible actions which enable o, and therefore a different *outcome_probability*. To model this accurately and maintain readability, we shall first introduce some more helper functions. Let us begin by defining the set of possible future states and their likelihoods after an action a has been executed in w_n:

$$execution : Worldstate \times action \mapsto \{(Worldstate, probability)\}$$

We define this function as:

$$execution(w_n, a) = \{(w, p) \mid w \in application(w_n, mutations), \text{ where} \\ mutations \in P(a.effects) \ \wedge \ p = mutationprobability(a.effects, mutations)\}$$

The P function here denotes the powerset operator, creating a set of all possible sets of mutations that can occur as a result of the set of effects present in a. *mutationprobability* is another helper function, signifying the probability with which a set of effects will result in a given set of mutations.

$$mutationprobability : \{effect\} \times \{mutation\} \mapsto probability$$

We define *mutationprobability* as:

$$mutationprobability(effects, mutations) = achieved\_mutations(effects, mutations) * unachieved\_mutations(effects, mutations)$$

As we can see, *mutationprobability* defines two helper functions of its own, *achieved_mutations* and *unachieved_mutations*. These functions calculate the probability of the mutation that have and have not been created out of a set of effects.

$$achieved\_mutations : \{effect\} \times \{Mutation\} \mapsto probability$$

$$unachieved\_mutations : \{effect\} \times \{Mutation\} \mapsto probability$$

$$achieved\_mutations(effects, mutations) = \prod\{e.p \mid \forall e \in effects \text{ where } \exists m \in mutations \text{ where } e = m\}$$

$$unachieved\_mutations(effects, mutations) = \prod\{1 - e.p \mid \forall e \in effects \text{ where } \neg\exists m \in mutations \text{ where } e = m\}$$

Together, these functions allow *execution* to generate a set of possible world states and their likelihoods based on a start state and an action. When we combine *execution* with *outcome_probability*, we are finally able to make an estimate of the probability with which a certain outcome will be generated by *nature* after execution of an action in a specific world state:

$$action\_outcome\_probability : Worldstate \times Action \times mutation \mapsto probability$$

$$action\_outcome\_probability(w, a, o) = \sum\{outcome\_probability(w_e, o) * p \mid (w_e, p) \in execution(w, a)\}$$

As we can see, *action_outcome_probability* generates a set of possible world states following the execution of an action a. The likelihood of the outcome o being generated in each of these states is multiplied by the probability that the state itself will be generated. This creates a set of probabilities for o being generated after each possible execution of a. The sum of these probabilities is thus equal to the chance o will be generated despite (or because of) execution of a.

When we compare the likelihoods of o occurring before (using *outcome_probability*) or after execution of a (using *action_outcome_probability*), we can observe whether a is an action that really does block o. If o is less likely to occur after a has been executed in state w, we can say that a blocks o in w:

$$blocks : Worldstate \times action \times Mutation \mapsto boolean$$

$$blocks(w, a, o) = action\_outcome\_probability(w, a, o) < outcome\_probability(w, o)$$

blocking is an important concept because it is needed to allow our agents to evaluate factors like possibility and effort. We will read more about this in 5.3. First, we will define the negligent interval, which defines the domain of worldstates for our agents' reasoning.

5.2.2 The negligent Interval

Fundamental to the idea of negligence is that of a *bad outcome*. We have defined a bad outcome as a mutation that is perceived negatively by at least one agent. In our agent environment we can identify a bad outcome o for agent g as follows:

$$utility(g, o) < 0, \text{ where } o \in w_n.mutations$$

Every bad outcome o has a *negligent interval* or $NI(o)$ associated with it. The negligent interval is the interval of time during which o was possible to be produced by at least one action. It is represented by a set of continuous world states previous to the realization of o, where in every state o is possible. Searches for attributions of negligence to agents will be confined to this interval.

Before we identify the function that creates the negligent interval, we shall create a few helper functions. To find the world state associated with a mutation, we create the function:

$$statefrommutation : mutation \mapsto Worldstate$$

and define it as:

$$statefrommutation(m) = \{w_n \mid w_n \in W, \text{ where } m \in w_n.mutations\}$$

The *interval* function returns a set of states between two points in time:

$$interval : time \times time \mapsto \{Worldstate\}$$

which is defined as:

$$interval(t_a, t_b) = \{w_n | w_n \in W, t_b > w_n.time > t_a\}$$

These helper functions let us create the negligent interval as follows:

$$NI : mutation \mapsto \{Worldstate\}$$

which is defined as:

$$NI(o) = \{w_n \mid w_n \in W, \text{ where } possible(w_n, o) = true \wedge \forall w_k \in interval(w_n.time, statefrommutation(o).time) : possible(w_k, o) = true\}.$$

To put this in more human understandable terms: The negligent interval is the continuous set of states where o was able to occur prior to o occurring. Thus, non-contiguous sets of states where a certain outcome was possible will not be counted as a single negligent interval, since they have states dividing them in time in which it was not possible for this outcome to occur.

Example of a negligent interval:

This example is about a hospital and our agent g. First we will sketch the environment and the way it is encoded in our model. Secondly, we will evaluate the way in which the negligent interval is constructed.

In our hospital, power is provided by the local city power supply and a backup generator. The backup generator is only used when the main city power supply fails. When both of these power sources fail, the hospital is left without power and certain important functions fail.

In our model, this is encoded as follows: We start with the action *backup power supply failure*. This action happens at random about once a year and is executed by *nature*. When it is executed, it has an effect *backup power off*, changing variable *backup power* from *on* to *off*. Reparation of the backup power supply happens with *repair backup power supply*. This action is to be executed by g and has an effect *backup power restored* that changes *backup power* back to *on*. Another action called *total power failure* can only be executed when there is a backup power failure. Consequently, it has a precondition *backup power off*, that evaluates to true when variable *backup power* is equal to *off*.

Now, suppose the hospital has had a backup power failure last year, and is experiencing another one this year. We start the model last year, at state w_0. In this state, *backup power* is *on*, so we have nothing to fear. Even though these actions happen almost instantaniously, we still have to model them with a beginning and an end, so every action produces two new states; one at the beginning of the action when it is placed in the partial actions list, and one at the end, when it generates an effect. At the first power failure, the execution of *backup power failure* is initiated, putting this action in the partial actions list and thus causing w_1. Once it is completed, it causes power to fail and creates state w_2. In state w_2, it is possible for *total power failure* to be executed. However, backup power is restored and we are again two states ahead, in w_4. At the second power failure, we enter state w_6, and again, *total power failure* is possible. After a while, the city power fails, causing the action *total power failure* to be executed and the effect *power off* to be realized in state w_8. This effect changes variable *power* from *on* to *off* and has a negative utility of -1000 for our agent g.

We shall now construct the negligent interval for the negative outcome *power off*. We start by providing the state list, along with the evaluation of the precondition *backup power off* and the partial actions list:

State trace		
State	Backup power off	Partial actions
w_0	false	
w_1	false	Backup power supply failure
w_2	true	
w_3	true	Repair backup power supply
w_4	false	
w_5	false	Backup power supply failure
w_6	true	
w_7	true	Total power supply failure
w_8	true	

Application of the function *negligent interval(power off)* will now yield the set of all states before w_8 where *backup power off* evaluates to true and had no states between them and w_8 where *backup power off* was false: $\{w_6, w_7\}$. Note that the negligent interval does not include the state where the negative

outcome occurred. This is because it is not useful for our agent to reason about this state when the has already happened.

The concept of a negligent interval is an important component in our agent framework, because it specifies the domain of world states that our agent will consider when making attributions about negligence. Now that we have specified how our agent composes this interval, we are ready to discuss how it evaluates the factors in our model for each of the states in it. This is the subject of the next section.

5.3 Evaluation of Factors

The attributional model presented in chapter four makes use of a number of factors (possibility, effort, intention, excuses, *etc.*) that are used to make appropriate inferences for the attribution of negligence and emotional arousal. These factors form the core of our framework, and after having defined the agent environment and some basic reasoning skills, it is now possible to develop an algorithmic interpretation for the evaluation of these factors, which is what we will do in this chapter.

5.3.1 Overview

In this section we will give an overview of how an agent can construct an opinion about another agent g regarding a negative outcome o. The end-result of such a computation is an opinion about g as defined in 5.1, consisting of a negligence rating and a set of emotions.

5.3.2 Context

The computations about to be described are not specific to either a negligent agent, or an agent suffering from another's negligence. Rather, the computations themselves remain largely agnostic to the perspective of which agent the self is, except for the generation of emotional responses. The only thing that is important here, is that the observing agent that we might refer to as the *observer*, possesses a to the *observer* seemingly complete world view.

What we mean by this is that the *observer* must have an idea of the state of relevant variables to be able to form an opinion about a certain agent g. An example would be that in our goldfish scenarios, the *observer* must be aware of g's location and the state of the fish. If the *observer* lacks these kinds of information, it is not possible to form an opinion of negligence.

Our algorithm starts by constructing the negligent interval $NI(o)$. After $NI(o)$ is constructed, a number of steps are taken for each state to form an evaluation about the behavior of g. These steps include checking if g possessed the possibility of blocking o, if g could have been expected to have made a stronger effort to prevent o, if g had an excuse, and to what extent g should have foreseen that he was being negligent. Together, these factors allow an agent to form an evaluation about g's behavior in this state.

The next few sections will describe in detail how the various evaluation steps mentioned above for a negative outcome o can be applied to a single agent g in a single state w. Followed by this is a description of how we take these states and combine them to form a single evaluation of g in state w_n, consisting of a negligence judgment and a set of related emotions. Finally, we present a method to aggregate the evaluations from multiple states in a single negligent interval into an opinion about g, which can then be added to g's representation in our agent's world view.

5.3.3 Possibility

The first factor in our evaluation of the attribution of negligence is that of *possibility*. Possibility is the determination of whether it was *possible* for g to do anything about o from occurring. This issue is split up in two questions:

1. Did the agent have foreknowledge of o?

2. If so, could the agent have done something to avoid o from happening?

Having *foreknowledge* of o means the agent knew about a way that o could have occurred. For this to hold true, g needs two things: An action a that could cause o. Secondly, g needs to be able to inspect the preconditions for a to make sure a is possible. Of course, g can only do this when all variables involved in the preconditions for a are visible to g:

$$preconditions\_visible : Worldstate \times agent \times action \mapsto boolean$$

We define *preconditions_visible* as:

$$preconditions\_visible(w, g, a) = true \Leftrightarrow \forall c \in a.preconditions : visible(w, g, c.variable)$$

preconditions_visible returns true only when all variables needed for the evaluation of the preconditions for a are visible to g. The answer to our first question can then be given by a new function, $foreknowledge(w, g, o)$:

$$foreknowledge : Worldstate \times agent \times mutation \mapsto \{action\}$$

$$foreknowledge(w, g, o) = \{a \mid a \in causesof(o) \wedge preconditions\_visible(w, g, a) \wedge possible(w, a)\}$$

The function $foreknowledge$ returns the set of actions that g knows are possible and that can cause o. If this is an empty set, we already know that g did not know about the possibility of o occurring. If the set is not empty however, g knows o can occur through any element in $foreknowledge(w, g, o)$. We shall now look at the second question, which asks what g can do about the possibility of o occurring. This is done by creating a new function, *ability*:

$$ability : Worldstate \times agent \times mutation \mapsto \{action\}$$

$$ability(w, g, o) = \{a \mid a \in actions, \text{ where } blocks(w, a, o) = true \wedge a.agent = g\}$$

ability returns a set of actions that g can execute to block o. If *ability* equals the empty set, g cannot do anything about o occurring. Having answered our two subquestions, we can now define the function *possibility*:

$$possibility : Worldstate \times agent \times outcome \mapsto boolean$$

$$possibility(w, g, o) = true \Leftrightarrow |foreknowledge(w, g, o)| > 0 \wedge |ability(w, g, o)| > 0$$

According to the definition above, *possibility* returns true when g both knows o can occur and g is in the position to do something about it. In the next section, we shall look at when an agent has made enough of an effort to stop o from occurring.

5.3.4 Effort

In this section, we look at the *effort* an agent has taken towards trying to prevent o in state w_n. There are two ways an agent can be deemed to have made a sufficient effort: The agent was busy trying to block o in state w_n or the agent started to block o soon after this state as a result deliberating for a while. Using the above information, we have split this factor up in two subquestions:

1. Was g trying to block o in w_n?
2. Did g initiate a blocking action a within the deliberation time $g.delib$?

We will answer subquestion number one by looking at the partial actions set $w_n.partials$ to find if there is an action in which g tried to block o:

$$made\_an\_effort : Worldstate \times agent \times mutation \mapsto boolean$$

$$made\_an\_effort(w, g, o) = true \Leftrightarrow \exists a \in w_n.partials : a.agent = g \wedge blocks(w, a, o)$$

$made\_an\_effort$ returns true when there is an action in the partial actions set that is being executed by g and blocks o. If $made\_an\_effort$ returned false, it is still possible that g was still deliberating its options and started to try to block o somewhere nearby in the future. If this is true, then there exists a state where g is trying to block o named w_k which satisfies $w_n.time < w_k.time <= w_n.time + g.delib$.

$$started\_an\_effort : Worldstate \times agent \times mutation \mapsto boolean$$

$$started\_an\_effort(w_n, g, o) = true \Leftrightarrow \exists w_k \in NI(o) \wedge w_n.time < w_k.time <= w_n.time + g.delib \wedge made\_an\_effort(w_k, g, o)$$

If $started\_an\_effort$ returned false too, then there was no action that g has undertaken before or on w_n or will undertake after w_n to stop o from happening. In this case, effort is false.

$$effort : Worldstate \times agent \times mutation \mapsto boolean$$

$$effort(w, g, o) = made\_an\_effort(w, g, o) \vee started\_an\_effort(w_n, g, o)$$

5.3.5 Excuses

An excuse is something the agent was doing during the negligent interval that was not related to the outcome o, but that the agent was doing for a different reason. This reason might be to increase its own utility, that of another agent, or even to help the agent that was suffering from the negligent outcome in a different way. Excuses can be divided into good excuses and bad excuses. We distinguish between a good and a bad excuse by looking at the size of the expected utility compared to the expected loss of utility because of negligence. A good excuse will gain more total utility than the expected loss of utility due to o will be, and a bad excuse will gain less.

The first thing we need is a helper function that gives us the expected utility of an action:

$$expected\_utility : Worldstate \times agent \times action \mapsto utilityrating$$

$$expected\_utility(w, g, a) = \sum\{utility(w, g, e) \mid e \in a.effects\}$$

Next we want to know the aggregate utility for all agents for this action:

$$total\_expected\_utility : Worldstate \times action \mapsto utilityrating$$

$$total\_expected\_utility(w, a) = \sum expected\_utility(w, g, a),\ \forall g \in G$$

Where G is our usual collection of agents. We also want to know the total utility for all agents for the negative outcome:

$$total\_utility : mutation \mapsto utilityrating$$

$$total\_utility(m) = \sum utility(g, m), \forall g \in G$$

Now that we have our utility functions, we are able to make an estimate of the utility of both any action a and any negative outcome o occurring in w. Better still, we can compare them to each other. But instead of using addition, we use subtraction to see whether g's action a really expects to bring more positive utility into the world than o could take out:

$$positive\_action : Worldstate \times action \times mutation \mapsto utilityrating$$

$$positive\_action(w, a, o) = total\_expected\_utility(w, a) - total\_utility(o)$$

What this tells us is that when *positive_action* is positive, g had a strong excuse, because he was doing something very good (at least in our agent's eyes). But when *positive_action* is negative, g could have better been helping to stop o instead of trying to do something which was not so important.

Because we must distinguish between good and bad excuses, the final functions for this factor will be two: *good_excuse*, and *bad_excuse*:

$$good\_excuse : Worldstate \times agent \times mutation \mapsto boolean$$

$$bad\_excuse : Worldstate \times agent \times mutation \mapsto boolean$$

Their definitions are given below:

$$good\_excuse(w, g, o) = true \Leftrightarrow \exists a \in w.partials : \neg blocks(w, a, o) \wedge a.agent = g \wedge positive\_action(w, a, o) >= 0$$

$$bad\_excuse(w, g, o) = true \Leftrightarrow \exists a \in w.partials : \neg blocks(w, a, o) \wedge a.agent = g \wedge positive\_action(w, a, o) < 0$$

5.3.6 Certainty

The factor of certainty represents the degree to which an agent knows that intervention on the part of the agent is required to avoid letting a bad outcome be established. In other words, when an agent is certain, it knows that to avoid letting o happen, the agent must do something, whereas when the agent is not certain (or uncertain), there is a realistic chance that o might not happen and the conditions which enable o to be generated might well change to make o impossible.

In our model, this means that g is able to block o, but there is no action from *nature* which can block o. Before we define our certainty function, we will define a helper function *agentprevents*:

$$agentprevents : Worldstate \times agent \times mutation \mapsto boolean$$

$$agentprevents(w, g, o) = true \Leftrightarrow \exists a \in actions\ where\ a.agent = g \wedge \exists\ (s, p) \in execution(w, a) :\ \neg possible(s, o)$$

The function *agentprevents* describes whether it's possible for an agent to execute an action a in world state w whereby the result can be such that o is not possible, thereby preventing o from occurring. This allows us to construct our *certainty* function in a simple manner:

$$certainty : Worldstate \times agent \times mutation \mapsto boolean$$

$$certainty(w, g, o) = true \Leftrightarrow agentprevents(w, g, o) \wedge \neg agentprevents(w, nature, o)$$

The *certainty* function is defined rather straightforward using the *agentprevents* function. Only when g is able to prevent o from happening and *nature* clearly isn't will *certainty* evaluate to true for g.

5.3.7 Intention

Intention is the easiest factor to determine, because we can simply use the agent's utility function to check whether the outcome has a positive utility for the agent:

$$intention(g, o) = utility(g, o) > 0$$

This is the last factor in our model. In the next section, we will see how these factors can combine together to form an evaluation of negligence.

5.4 Generating evaluations

By now our agent has done a great deal of reasoning, and is almost ready to provide us with an evaluation of what it thinks of its fellow agents behavior. For a particular negative outcome o, for a particular agent g, in a particular world state w_n, it has deliberated whether g possessed the *possibility* of preventing o. It has evaluated whether g had made a good *effort* at trying to block o, or had a good or a bad *excuse* for letting o happen. Finally, it has also evaluated whether g was aware that g alone could have prevented o from happening, and whether g might even have *intended* o to happen.

The algorithm we are about to present is based on the results of the questionnaire in chapter four. It is split in two parts: In the first, *evaluate_negligence*, we will use the evaluations of the various factors to come to a negligence rating. In the second part, we will use the negligence rating combined with the factor intention to select an emotional response, not necessarily only to g, but perhaps to other agents as well. The result of this calculation will consist of a set of evaluations, where an evaluation consists of a negligence rating together with an emotional response. We start by providing the prototype of *evaluate_negligence*:

$$evaluate\_negligence : Worldstate \times mutation \times agent \mapsto negligence\_rating$$

The nature of this function is somewhat different than that of our other functions, so we will step away from our usual mathematical, set-oriented notation, and use a more imperative, pseudo-code notation this time.

```
evaluate_negligence(w,g,o) =
{
    if(not possibility(w,g,o))
        negligence_rating = 0 (Minimal);
    if(effort(w,g,o) or good_excuse(w,g,o))
        negligence_rating = 1 (Low);
    if(bad_excuse(w,g,o) or certainty(w,g,o))
        negligence_rating = 5 (High);
    else
        negligence_rating = 3 (Medium);
    return negligence_rating;
}
```

For the second part of our algorithm, we will be attaching emotional evaluations to these negligence ratings. Let an evaluation be defined as:

$$evaluation = (mutation, state, agent, negligencerating, emotions)$$

We can then define the second part of our evaluation function:

$$evaluate\_state : Worldstate \times agent \times mutation \mapsto \{evaluation\}$$

Again, because of the nature of the function, we will use pseudo-code this time. Comments start after a pound (#) sign .

```
evaluate_state(w,g,o) =
{
    let eval be an empty set of evaluations
    let em be an empty set of emotions
    let em2 be an empty set of emotions

    negligence = evaluate_negligence(w,g,o)

    if(g = i)
    {
        # if we are the agent
```

```
    # If we are negligent, feel fear, guilt and shame accordingly
    if negligence > 1
    {
        add (Fear, negligence / 3) to em
        add (Guilt, negligence / 2) to em
        add (Shame, negligence / 2) to em
    }

    # Happiness and sadness are generated according to our utility function.
    if intention(w,g,m) = true
    {
        add (Happiness, 1) to em
    }
    if intention(w,g,m) = false
    {
        add (Sadness, 3) to em
        add (Sympathy, 3) to em
    }

    # attach this evaluation to every agent who suffered from my negligence
    forall agents h in G: if intention(w,h,m) = false
    {
        add (o, w, h, 0, em) to eval
    }
} else
    # if the agent under scrutiny is someone else

    # If he is negligent, we are angry at him
    if negligence > 1
    {
        add (Anger, 4) to em
    }

    # If he did not intend it, we are sad for him and have sympathy
    if intention(w,g,o) = false
    {
        add (Sadness, 2) to em
        add (Sympathy, 1) to em
    }
    add (o, w, g, negligence, em) to eval

    # Feel sadness and sympathy for any agents that did not intend this.
    if intention(w,g,m) = false
    {
        add (Sadness, 3) to em2
        add (Sympathy, 3) to em2
    }
    # attach this evaluation to every agent who suffered from g's negligence.
    forall agents h in G: if intention(w,h,m) = false
```

```
        {
            add (o, w, h, 0, em2) to eval
        }
    }
    return eval;
}
```

The *evaluate_state* function describes the way emotional responses are generated based on a negligence rating and the intentions of various agents. When the agent turns out to be negligent itself, it feels shame and guilt towards all agents that suffered from the negligent action.

When it is evaluating another agent and this agent turns out negligent, it becomes angry with this agent. In the next section, we will see how the result of *evaluate_state*, a set of evaluations, can be combined to form opinions about other agents, and about oneself.

5.5 Putting it all together

In the previous sections we have seen how to evaluate various factors in a certain world state and come to an evaluation regarding a set of agents. However, a negligent interval can quite often be composed of more than one state. In this section, we will see how we can take a set of evaluations about a set of agents and transform these into a set of opinions about these agents, rendering a final verdict of negligence and emotional response.

We deal with multiple evaluations in our model by calculating the agent's opinion as a weighted average of the evaluations of all states in the negligent interval. Since an evaluation for a state that existed for hours is most likely more important than an evaluation for another state in the same negligent interval that only existed for a few minutes, we regard the first evaluation as more important and use the states' existence lengths as weights in the calculation of the average evaluation.

The time a state has existed can be measured by the time indices between that state and it's successor. Let the function *statetime* indicate the time the state existed:

$$statetime : Worldstate \mapsto time$$

Let *totaltime* be the set-equivalent of this *statetime*

$$totaltime : \{Worldstate\} \mapsto time$$

and be defined as:

$$totaltime(states) = \sum\{statetime(s) \mid \forall\, s \in states\}$$

Using *statetime* and *totaltime* we can integrate a set of evaluations about a single agent into an opinion about that agent for the entire negligent interval:

$$integrate : \{evaluation\} \mapsto opinion$$

The negligence rating as well as the emotions in this opinion are a time-weighted average of the set of states.

The last function we will define in this thesis is called *evaluate_agent*. It evaluates the actions of an agent g over an entire negligent interval, and captures all the resulting evaluations from the observing agent. It then uses *integrate* to let the observing agent form opinions about the behavior of g during the negligent interval.

$$evaluate\_agent : agent \times outcome$$

```
evaluate_agent(g,o) =
{
    let eval be an empty set of evaluations

    construct the negligent interval NI(o)

    forall states s in NI(o)
    {
        # here we collect the evaluations our agent makes
        # regarding all affected agents in eval
        add evaluate_state(s,g,o) to eval
    }

    forall agents g in G
    {
        # eval may contain evaluations for many agents,
        # handle them one by one
        take the subset gs of evaluations about g from eval

        # we form an opinion for g after integrating
        # all evaluations from all different states
        # if we have evaluations about g, turn them into opinions.
        if(|gs| > 0)
        {
            opinion op = integrate(gs)

            # we add our opinion to g
            add(g, opinion)
        }
    }
}
```

5.6 Discussion

In this chapter we have introduced our agent model and shown how it can be used to let an agent form opinions about the behavior of other agents, including itself, during an interval of time that led up to a negative outcome. The computational model we have created is able to capture a decent subset of human reasoning about negligence attributions, but it is not perfect:

Humans can attribute negligence based on broken agreements between each other, but currently our model is not able to do this. For it to be able to

incorporate these features, it would need to know about speech act theory, and understand the concept of commitments.

Furthermore, an agent can still keep itself out of harms way by not knowing about its environment. If an agent sticks its head in the sand, it has no view of the world and cannot be blamed about being negligent. This is similar to the problems encountered in Mao's model, but less severe, since in our model an agent will have to actively avoid contact to be judged non-negligent and can not just stand on the sidelines.

Another area for improvement would be the inclusion of a social norms system. Currently, our agents monitor aggregate utility as the prime good by which to judge the actions of another agent, much like the learned hand rule. The central idea behind the learned hand rule is not likely to need adapting, but it is not improbable that humans might value certain sets of variables as having different utility based on context.

6 Evaluation

In the previous chapter, we have defined our agent environment and explained how it is able to reason about negligence. It is able to consider a set of previous world states, deliberate, and form a set of opinions regarding an agent that could have prevented the negative outcome, as well as the agents that are suffering from this outcome.

In this chapter, we are going to evaluate whether our model matches the results we have obtained from our questionnaire in chapter four. We start by providing a step-by-step trace for the evaluation process for the first scenario. The results of this computation are discussed and then compared to the human data.

Evaluation of our model can be done in tree steps: First, we describe how we encode the scenario from the questionnaire into our agent environment. Second, we analyze the computations performed for our agent to reach a result. Lastly, we compare the results of our agent framework to that of the questionnaire.

6.1 Initializing the agent environment

Let us begin by defining our agent environment by looking at our agents themselves. Our agents' names are *Andre* and *Bob*. *Andre* and *Bob* share an apartment, but are able to leave as well. This makes it a wise choice to model their location as variables. The center of attention for the questionnaire is Andre's goldfish. The goldfish's health goes through a number of states and is a key element to our attribution process, so that is going to be a variable as well. The following table shows these variables along with their initial values:

Variables	
Name	Value
Location Andre	home
Location Bob	home
behavior goldfish	normal
Health goldfish	healthy
water conditions	normal

Of course, things are not going to stay this way. We need to define a set of conditions that we can use to define actions and their preconditions.

Conditions			
Name	Variable	Relation	Value
Andre home	location Andre	equals	home
Bob home	location Bob	equals	home
fish healthy	behavior goldfish	equals	normal
fish weird	behavior goldfish	equals	weird
fish sick	health goldfish	equals	sick
water cleaned	water conditions	equals	clean
water normal	water conditions	equals	normal

Next, we shall determine the actions that our agents are able to execute. We start with the center of attention, the goldfish. Since the fish is not able to execute any actions by itself, it sometimes gets sick. We model the transitions between the various health states of the fish in the following way:

When the fish is healthy, it is able to get a little weird. In this case, the fish can be expected to act weird too, but this is in itself nothing serious. The fish can recover from this by having *nature* execute the action *recover from weirdness*, or it can die from behaving unnaturally by having *nature* execute the action *die from weirdness*. The fish is also able to become sick. This is a far more serious ailment, as the chances that the fish will die are much greater. When the fish is sick, this means something is wrong, and either Bob or Andre will have to do something to save the fish from dying. Next, we will discuss what the agents are able to do. It is possible for both Andre and Bob to leave the apartment and come back. The actual length of their journey does not really matter much, because they tend to stay out for an entire weekend at a time anyway. When they are at home, however, they are able to feed the fish, and to clean its water. We encode these actions into our database as follows:

Our action database:

Actions			
Name	Agent	Precond.	Effects
Andre leaves	*Andre*	{Andre at home}	{Andre away}
Bob leaves	*Bob*	{Bob at home}	{Bob away}
Andre arrives home	*Andre*	{Andre away}	{Andre at home}
Bob arrives home	*Bob*	{Bob away}	{Bob at home}
Andre cleans fish water	*Andre*	{Andre at home}	{water clean}
Bob cleans fish water	*Bob*	{Bob at home}	{water clean}
fish gets weird	*nature*	{fish healthy}	{fish gets weird}
fish recovers from weirdness	*nature*	{fish weird}	{fish gets healthy}
fish gets sick	*nature*	{fish healthy}	{fish gets sick}
fish recovers from sickness	*nature*	{fish sick}	{fish gets healthy}
fish dies 1	*nature*	{fish sick}	{fish dies}
fish dies 2	*nature*	{fish weird, water cleaned}	{fish dies}
fish dies 3	*nature*	{fish weird, water normal}	{fish dies}

Actions, continued			
Name	duration	visible	likelihood
Andre leaves	1 hour	true	1.0
Bob leaves	1 hour	true	1.0
Andre arrives home	1 hour	true	1.0
Bob arrives home	1 hour	true	1.0
Andre cleans fish water	1 hour	true	1.0
Bob cleans fish water	1 hour	true	1.0
fish gets weird	5 hours	false	0.02
fish recovers from weirdness	5 hours	false	0.50
fish gets sick	5 hours	false	0.02
fish recovers from sickness	5 hours	false	0.10
fish dies 1	3 hours	false	0.40
fish dies 2	3 hours	false	0.04
fish dies 3	3 hours	false	0.20

The preconditions and effects tables that accompany this actions database are given below:

preconditions			
Name	variable	relation	value
Andre at home	location Andre	equals	home
Andre away	location Andre	does not equal	home
Bob at home	location Bob	equals	home
Bob away	location Bob	does not equal	home
Fish healthy	behavior goldfish	equals	healthy
Fish weird	behavior goldfish	equals	weird
Fish sick	behavior goldfish	equals	sick

effects			
Name	variable	value	p
Andre at home	location Andre	home	1.0
Bob at home	location Bob	home	1.0
Andre away	location Andre	away	1.0
Bob away	location Bob	away	1.0
fish gets healthy	behavior fish	healthy	1.0
fish gets weird	behavior fish	weird	1.0
fish gets sick	behavior fish	sick	1.0
fish dies	health fish	dead	1.0

Now, we need to define the utility functions for *Andre* and *Bob*. For this scenario, we will give them an identical utility function $utility_1$:

$utility_1$		
Name	value	utility
behavior goldfish	normal	100
behavior goldfish	weird	90
health goldfish	healthy	1000
health goldfish	dead	0

Having defined our agent environment for the simulation of scenario one, we can now proceed to the second stage of our trace, where we trace through the reasoning steps required for our agent to form its opinion about these agents.

6.2 Computational analysis

In this section, we will retrace the steps needed for our agent to make its attributions of negligence. We start by giving a concise description of what has happened in our scenario:

First, Andre and Bob are home alone. Then, Andre leaves, leaving Bob alone with the fish. Bob notices the fish acting weird, but doesn't do anything, probably thinking the fish will revert back to normal any time. However, the fish does not recover from its weirdness, and dies.

We can get a visual overview of the situation by providing a state trace, along with the value of the precondition *fish weird* and any partial actions:

state trace for scenario$_1$		
State	*fish weird*	partial actions
w_0	false	$\emptyset$
w_1	false	$\{Andre\ leaves\}$
w_2	false	$\emptyset$
w_3	false	$\{fish\ gets\ weird\}$
w_4	true	$\emptyset$
w_5	true	$\{fish\ dies\ 3\}$
w_6	true	$\emptyset$

Again, any action that is initiated and completed triggers the creation of two states: One for creation of the action and the other one for completion. What we are interested in now, is what our agent thinks of the behavior of *Bob*, since he seemed to be able to do something about the fish dying. To achieve this, we let our agent execute the function *evaluate_agent*:

Now the time has come for us to introduce a new concept, namely that of the *execution trace*. An execution trace is similar to a state trace, except that it traces the execution of functions. An execution trace consists of three columns, providing us with an overview of what is happening while the agent is deliberating. The first column mentions the currently executing function, including its arguments. The second column mentions the function that is being called inside the function in the first column. Finally, the third column returns the result of the execution of the function in the second column. The bottom line shows the result of the function itself. As an example, we will give an execution trace of the function *possible*. This function is executed in state w_0 of our previous state trace and concerns the action *Bob leaves*.

execution trace for possible()		
Function	subfunction	result
$possible(w_0, Bob\ leaves)$	$evaluate(w_0, Bob\ at\ home)$	true
$possible(w_0, Bob\ leaves)$		true

In this case, we have chosen a rather simple function, as *possible* only needs to evaluate one subfunction, *evaluate*. As *evaluate* returns true, so does *possible*. On the following page, we have provided a partial execution trace for the function *evaluate_agent*, where $g = Bob$ and $o = fish\ dies$:

The table provides a chronological function call history, allowing us to trace the execution of *evaluate_agent* through its subfunctions. The negligent interval is constructed at, #3, and yields two states: w_5 and w_6. These states are evaluated at #3 and #37, diving deeper into their subfunctions and arriving at the attribution factors of *possibility* at #4, followed by *effort* at #4, good and bad *excuses* at #24 and #25, and finally *certainty* at #26.

The trace is partial because some of the lower-level functions such as *negligent interval* and *blocks* are not entered in full depth. Also, #38 is not expanded because the only difference with state #4 is a partially executed action by *nature* that lets the fish die *fish dies 3*. This action has no effect on the underlying calculation, and therefore we use the same return value as calculated in #4. The final result of this calculation are two opinions for *Andre* and *Bob*, as can be seen in the execution trace and is summarized here in tabular form again:

Final results for simulation of scenario 1:

Agent	mutation	negligence rating	emotions
Andre	fish dies 3	0 (Minimal)	{(sadness, 3), (sympathy, 3)}
Bob	fish dies 3	3 (Medium)	{(anger, 4), (sadness, 2), (sympathy, 1)}

6.3 Comparison and discussion

When we compare this to the results of our questionnaire, we see that our model has made a correct estimation of both the negligence rating as well as the emotional arousal.

Questionnaire results for scenario 1:

Agent	negligence rating	emotions
Andre	0 (Minimal)	{(sadness, 2.8), (sympathy, 2.6)}
Bob	3.4 (Medium)	{(anger, 3.0), (sadness, 2.1), (sympathy, 0.8)}

The main differences in these results are the anger ratings. This is due to the fact that human anger ratings vary more per scenario and our model only uses a default value of three (medium) when an agent is deemed sufficiently negligent. Parameterizing our anger ratings according to the negligence rating and other circumstances is likely to improve this scenario, but this falls outside of the scope of this research. In the next chapter, we will draw conclusions and give suggestions for future research.

	Function	subfunction	result
#1	$evaluate\_agent(Bob, fishdies)$	$NI(fishdies)$	$\{w_5, w_6\}$
#2	$evaluate\_agent(Bob, fishdies)$	$add\_evaluate\_state(w_5, Bob, fishdies)$	
#3	$evaluate\_state(w_5, Bob, fishdies)$	$evaluate\_negligence(w_5, Bob, fishdies)$	$\{(fishdies, w_5, Bob, 3,$ $\{(Anger, 4), (Sadness, 2), (Sympathy, 1)\}),$ $(fishdies, w_5, Andre, 0, \}$ $\{(Sadness, 2), (Sympathy, 3)\})\}$
#4	$evaluate\_negligence(w_5, Bob, fishdies)$	$possibility(w_5, Bob, fishdies)$	3 (Medium)
#5	$possibility(w_5, Bob, fishdies)$	$foreknowledge(w_5, Bob, fishdies)$	{ fish dies 3 }
#6	$foreknowledge(w_5.Bob, fishdies)$	$causesof(fishdies)$	{ fish dies 1, fish dies 2, fish dies 3 }
#7	$foreknowledge(w_5.Bob, fishdies)$	$preconditions\_visible(w_5, Bob, fishdies1)$	true
#8	$foreknowledge(w_5.Bob, fishdies)$	$possible(w_5, fishdies1)$	false
#9	$possible(w_5, fishdies1)$	$evaluate(w_5, fishsick)$	false
#10	$foreknowledge(w_5.Bob, fishdies)$	$preconditions\_visible(w_5, Bob, fishdies2)$	true
#11	$foreknowledge(w_5.Bob, fishdies)$	$possible(w_5, fishdies2)$	false
#12	$possible(w_5, fishdies2)$	$evaluate(w_5, fishweird)$	true
#13	$possible(w_5, fishdies2)$	$evaluate(w_5, watercleaned)$	false
#14	$foreknowledge(w_5.Bob, fishdies)$	$preconditions\_visible(w_5, Bob, fishdies3)$	true
#15	$foreknowledge(w_5.Bob, fishdies)$	$possible(w_5, fishdies3)$	true
#16	$possible(w_5, fishdies3)$	$evaluate(w_5, fishweird)$	true
#17	$possible(w_5, fishdies3)$	$evaluate(w_5, waternormal)$	true
#18	$possibility(w_5, Bob, fishdies)$	$ability(w_5, Bob, fishdies)$	{ Bob cleans fish water }
#29	$ability(w_5, Bob, fishdies)$	$blocks(w_5, Bobcleansfishwater, fishdies)$	true
#20	$blocks(w_5, Bobcleansfishwater, fishdies)$	$action\_outcome\_probability(w_5, Bobcleansfishwater, fishdies3)$	true
#21	$evaluate\_negligence(w_5, Bob, fishdies)$	$effort(w_5, Bob, fishdies)$	false
#22	$effort(w_5, Bob, fishdies)$	$made\_an\_effort(w_5, Bob, fishdies)$	false
#23	$effort(w_5, Bob, fishdies)$	$started\_an\_effort(w_5, Bob, fishdies)$	false
#24	$evaluate\_negligence(w_5, Bob, fishdies)$	$good\_excuse(w_5, Bob, fishdies)$	false
#25	$evaluate\_negligence(w_5, Bob, fishdies)$	$bad\_excuse(w_5, Bob, fishdies)$	false
#26	$evaluate\_negligence(w_5, Bob, fishdies)$	$certainty(w_5, Bob, fishdies)$	false
#27	$certainty(w_5, Bob, fishdies)$	$agentprevents(w_5, Bob, fishdies)$	true
#28	$certainty(w_5, Bob, fishdies)$	$agentprevents(w_5, nature, fishdies)$	false
#29	$evaluate\_state(w_5, Bob, fishdies)$	$add(Anger, 4)toem$	
#30	$evaluate\_state(w_5, Bob, fishdies)$	$add(Sadness, 2)toem$	
#31	$evaluate\_state(w_5, Bob, fishdies)$	$add(Sympathy, 3)toem$	
#32	$evaluate\_state(w_5, Bob, fishdies)$	$add(fishdies, w_5, Bob, 3, em)toeval$	
#33	$evaluate\_state(w_5, Bob, fishdies)$	$add(Sadness, 3)toem2$	
#34	$evaluate\_state(w_5, Bob, fishdies)$	$add(Sympathy, 3)toem2$	
#35	$evaluate\_state(w_5, Bob, fishdies)$	$add(fishdies, w_5, Andre, 0, em2)toeval$	
#36	$evaluate\_agent(Bob, fishdies)$	$add\_evaluate\_state(w_6, Bob, fishdies)$	
#37	$evaluate\_state(w_6, Bob, fishdies)$		$Sameas\#3*$
#38	$evaluate\_agent(Bob, fishdies)$	$integrate(\{\#3, \#37\})$	$(fishdies, 3, \{(Anger, 4),$ $(Sadness, 2), (Sympathy, 1)\})$
#39	$evaluate\_agent(Bob, fishdies)$	$add(Bob, \#38)$	
#40	$evaluate\_agent(Bob, fishdies)$	$integrate(\{\#3, \#37\})$	$(fishdies, 0,$ $\{(Sadness, 2), (Sympathy, 3)\})$
#42	$evaluate\_agent(Bob, fishdies)$	$add(Andre, \#40)$	
#43	$evaluate\_agent(Bob, fishdies)$		#38 and #40

7 Conclusions and future research

After having compared our model against the human data in the previous chapter, we draw conclusions about our model and will end this chapter with suggestions for further research.

7.1 Conclusions

In this thesis, we have made a successful first attempt at creating an agent model capable of incorporating human attributions about negligence. The model is based on psychological literature pertaining to attribution theory. Though there is not a substantial amount of literature regarding negligence specific studies on the subject, our survey results were sufficient: A model was formulated that could be evaluated successfully against a human population by means of a field study. The model we have created is only partially based on both Shaver's theory and Mao's model; We have taken the staged approach of attribution theory, and kept some of its factors, such as foreknowledge and intention. We have then added negligence-specific factors such as lack of possibility and excuses, two items that are capable of mitigating an attribution of negligence.

The resulting model places the following demands on an agent environment: Agents are expected to have a reasonable amount of knowledge concerning possible actions that they and other agents are able to undertake. They must also be able to perform hindsight reasoning about expected utility of actions in an arbitrary world state. Furthermore, agents must be able to estimate the utility of *not* performing an action in a certain situation, having knowledge about a probable chain of events.

7.2 Further research

A few directions for further research have already been uncovered in chapters five and six: Integration of speech act theory could allow understanding of the concept of commitments. This is an important step towards letting an agent be responsible for agreements and contracts.

Developing an aspect of social norms would allow an agent to become sensitive to understanding the relative weight of negligent acts in a certain environment. This is analogous to the concept of *Duty of care* in legal research concerning negligence. Such a framework would allow agents to establish role-specific negligence ratings: A role-specific agent such as a medical agent can then be attributed more or less negligence based on the social norms and obligations relevant to a certain class of agent.

References

[**Austin 1962**] J.L. Austin, How to do things with Words: The William James Lectures delivered at Harvard University in 1955. Ed. J. O. Urmson. Oxford: Clarendon, 1962. ISBN 0674411528

[**Bratman 1987**] M.E. Bratman. Intention, Plans, and Practical Reason. Harvard University Press, 1987.

[**Bredeweg 2004**] Bert Bredeweg , Ken Forbus, Qualitative modeling in education, AI Magazine, v.24 n.4, p.35-46, January 2004

[**Brown 1973**] John Prather Brown, The Journal of Legal Studies, Vol. 2, No. 2. (Jun., 1973), pp. 323-349.

[**Codd 1970**] Codd, E. F. 1970. A relational model of data for large shared data banks. Commun. ACM 13, 6 (Jun. 1970), 377-387.

[**Feigenson 1997**] Feigenson, N., Park, J., & Salovey, P. (1997). Effect of blameworthiness and outcome severity on attributions of responsibility and damage awards in comparative negligence cases. Law and Human Behavior, 21,597-617.

[**Feigenson 2001**] Feigenson , N., Park, J., & Salovey, P. The role of emotions in comparative negligence judgments. Journal of Applied and Social Psychology, 31, 576-603, 2001

[**Feldman 1998**] Feldman, Allan M. & Frost, John M., 1998. "A simple model of efficient tort liability rules," International Review of Law and Economics, Elsevier, vol. 18(2), pages 201-215, June.

[**Fink 1996**] Pamela K. Fink , L. Tandy Herren, Modeling disease processes for drug development: bridging the gap between quantitative and heuristic models, Proceedings of the 28th conference on Winter simulation, p.1183-1190, December 08-11, 1996, Coronado, California, United States

[**Forbus 1984**] Forbus, K. Qualitative Process Theory. Artificial Intelligence, 24, 85-168, 1984

[**Gilles 2002**] Gilles, Stephen G., "The Emergence of Cost-Benefit Balancing in English Negligence Law" . Chicago-Kent Law Review, Vol. 77, No. 3, 2002, Forthcoming Available at SSRN: http://ssrn.com/abstract=315459 or DOI: 10.2139/ssrn.315459

[**Grady 1983**] Mark F. Grady, A New Positive Economic Theory of Negligence, The Yale Law Journal, Vol. 92, No. 5. (Apr., 1983), pp. 799-829.

[**Grady 1989**] Mark F. Grady, Untaken precautions, The Journal of Legal Studies, Vol. 18, No. 1. (Jan., 1989), pp. 139-156.

[**Gratch 2004**] J. Gratch, S. Marcella, A Domain-independent Framework for Modeling Emotion, Journal of Cognitive Systems Research, Volume 5, Issue 4, 2004, Pages 269-306

[**Heider 1958**] F. Heider. The Psychology of Interpersonal Relations. John Wiley & Sons Inc, 1958.

[**Knobe 2003**] J. Knobe. Intentional Action and Side-Effects in Ordinary Language. Analysis, 63:190-193, 2003

[**Lepper 1973**] Lepper, M. P., & Greene, D., & Nisbett, R. E., Undermining children's Intrinsic interest with extrinsic reward: A test of the "overjustification" hypothesis. JPSP, 1973, 28, 129-137.

[**Mao 2006**] W. Mao. Modeling Social Causality and Social Judgment in Multi-Agent Interactions. Ph.D. Dissertation. Computer Science Department, University of Southern California, 2006.

[**Noel 1987**] J. G. Noel, D. R. Forsyth, and K. N. Kelley, Improving the Performance of Failing Students by Overcoming Their Self-Serving Attributional Biases, Basic and applied psychology. 1987, 8(1 & 2), 151-162

[**Ortony 1990**] Ortony, A., & Turner, T. J. (1990). What's basic about basic emotions? Psychological Review, 97, 315-331.

[**Ortony 2001**] Ortony, A., On making believable emotional agents believable, Emotions in Humans and Artifacts, pages 189212. MIT Press, 2003.

[**SASO**] Stability and Support Operations Simulations and Training, ICT, Marina del Rey

[**Shaver 1975**] K. G. Shaver. An Introduction to Attribution Processes. Winthrop Publishers, 1975.

[**Shaver 1985**] K. G. Shaver. The Attribution Theory of Blame: Causality, Responsibility and Blameworthiness. Springer-Verlag, 1985.

[**Tomai 2007**] Tomai, E. and Forbus, K. (2007). Plenty of Blame to Go Around: A Qualitative Approach to Attribution of Moral Responsibility. In Proceedings of Qualitative Reasoning Workshop 2007, Aberystwyth, U.K.

[**Tracy 2006**] Tracy, J. L., & Robins, R. W. (2006). Appraisal antecedents of shame and guilt: Support for a theoretical model. Personality and Social Psychology Bulletin, 32, 1339-1351.

[**Weiner 1995**] B. Weiner. Judgments of Responsibility: A Foundation for a Theory of Social Conduct. The Guilford Press, 1995.

Appendix A - Questionnaire results

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
1	1	2	1	3,1,2,4	2	0	1	1	2	1	0	3	0	0	2	2	1	0
1	2	2	0	1,3,4	0	0	0	0	0	0	0	2	0	3	0	0	0	0
1	3	1	3	2	2	0	0	2	2	2	1	2	0	4	0	0	0	0
1	4	2	1	2,1,3	5	0	1	5	5	5	4	5	0	4	3	2	2	0
1	5	1	4	3,4,2,1	3	0	3	3	2	2	2	3	0	3	2	2	0	0
1	6	2	3	3,2	1	0	0	3	4	4	1	0	0	3	0	1	0	0
1	7	1	2	2,3,1,4	1	0	1	1	2	0	0	2	0	2	2	2	1	0
1	8	1	4	3,2,4,1	5	0	0	5	5	5	3	3	0	3	2	2	0	1
1	9	1	3	3	5	0	0	3	4	2	0	0	0	4	0	0	0	0
1	10	2	3	2,3	1	0	0	2	2	1	0	0	0	2	0	0	0	0
1	11	2	5	2,3	0	3	3	0	1	0	0	5	0	5	3	0	4	0
1	12	1	4	3,2,4,1	4	0	4	4	3	4	4	3	0	4	0	1	2	0
1	13	1	5	4,3,2	3	0	0	4	4	3	0	3	0	3	0	0	0	0
1	14	2	5	3	3	0	0	1	2	2	1	5	0	2	0	0	0	0
1	15	1	5	3,2	5	0	0	3	1	0	0	0	0	1	0	0	0	0
1	16	2	3	3,2	0	0	0	1	3	1	0	2	0	3	0	0	0	0
1	17	2	5	3,2	3	0	0	2	3	3	3	1	0	4	0	0	0	0
1	18	2	2	3	0	0	0	0	0	0	0	1	0	1	0	0	0	0
1	19	2	4	2,3,4	3	0	0	3	2	2	2	3	0	4	3	1	1	0
1	20	1	4	2	4	0	3	3	4	3	1	0	0	4	2	1	1	0
1	21	2	5	2,3	1	0	0	0	0	1	0	3	0	4	0	2	1	2
1	22	1	0	1,2	2	0	0	1	4	0	0	0	0	3	0	0	0	0
1	23	1	3	4,3,2,1	4	1	0	2	4	4	1	1	1	4	0	2	0	0
1	24	1	4	2,3,4,1	4	1	1	4	4	4	1	1	1	3	2	1	2	1
1	25	2	5	3	5	0	0	5	4	4	3	0	0	4	0	0	0	0
1	26	1	4	3	3	0	0	5	0	2	0	0	0	4	0	0	0	0
1	27	2	5	2,3	5	0	0	0	4	4	4	4	0	1	1	2	0	0
1	28	1	5	1,2,3,4	2	0	3	3	0	0	0	4	1	4	2	1	2	1
1	29	2	3	3,2,4	4	0	0	4	5	1	0	5	0	3	2	1	1	0
1	30	1	1	3,2,4,1	3	0	0	1	2	1	1	0	0	1	0	0	0	0
1	31	1	4	2,3	3	0	0	5	2	3	4	3	0	4	0	1	0	0

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
2	1	2	0	4,3,1,2,5	5	0	0	0	1	0	0	5	0	0	0	5	1	0
2	2	2	0	4,1,5	3	0	0	4	2	2	0	1	0	0	0	3	0	0
2	3	1	1	4	0	0	0	0	0	0	0	0	0	0	0	3	0	0
2	4	2	1	2,4,3	5	0	1	5	5	4	2	5	0	2	3	5	0	0
2	5	1	0	4,3	1	0	1	2	1	0	0	3	0	0	1	4	0	0
2	6	2	0	4,3,2	1	0	0	0	4	0	0	2	0	0	0	4	0	0
2	7	1	1	2,4,3,1,5	1	0	0	0	1	0	0	1	0	1	1	2	1	0
2	8	1	1	2,4	5	0	0	3	5	0	0	4	0	1	2	2	0	0
2	9	1	1	4	1	0	0	0	1	0	0	4	0	1	2	3	0	0
2	10	2	0	4	1	0	0	1	1	0	0	3	0	0	1	3	0	0
2	11	2	5	2,3	5	0	0	4	4	4	0	5	0	0	0	4	0	0
2	12	1	1	3,2,5,1,4	4	0	1	3	3	2	2	3	0	2	0	1	0	0
2	13	1	0	4,3,2	1	0	0	2	2	0	0	5	0	0	0	4	0	0
2	14	2	0	4	0	0	0	0	1	1	0	5	0	0	0	5	1	0
2	15	1	1	4,2	5	0	0	0	4	0	0	0	0	0	0	0	0	0
2	16	2	2	4,2	2	0	0	0	1	0	0	3	0	1	0	4	0	0
2	17	2	2	4,3,2	5	0	0	2	4	0	0	5	0	0	0	3	0	0
2	18	2	1	4	0	0	0	0	0	0	0	1	0	1	0	1	0	0
2	19	2	1	2,3,4,5	5	0	0	4	4	3	3	4	0	2	4	5	1	0
2	20	1	2	2,3	4	0	3	3	4	0	0	4	0	0	0	4	1	0
2	21	2	2	2,3,4	4	0	1	3	3	4	2	2	0	3	2	2	2	0
2	22	1	0	4,1,2	2	0	0	2	4	0	0	3	0	3	0	0	0	0
2	23	1	0	4,5,3,2,1	2	1	0	0	4	0	0	0	0	0	0	3	0	0
2	24	1	2	2,3,4,5,1	4	1	1	3	4	2	1	4	1	2	2	4	2	1
2	25	2	3	3,4	5	0	0	3	3	2	1	0	0	0	0	4	0	0
2	26	1	0	4	4	0	0	2	0	0	0	3	0	0	0	0	0	0
2	27	2	3	3,4	5	0	0	3	5	1	2	5	0	0	1	2	0	0
2	28	1	0	2,3,4,5,1	4	1	1	0	4	0	0	4	2	0	3	1	1	0
2	29	2	1	4,3	4	0	1	3	4	0	0	5	0	0	1	4	0	0
2	30	1	0	4,3,2,5,1	0	0	0	0	1	0	0	5	0	0	0	5	0	0
2	31	1	0	4	4	0	0	2	3	0	2	4	0	3	0	2	0	0

Scenario	Particpant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
3	1	2	4	4,3,5,2,1	3	0	0	4	3	4	1	4	0	4	1	0	1	0
3	2	2	5	4,1,5	0	0	0	0	0	0	0	0	0	4	0	0	0	0
3	3	1	4	4	2	0	0	2	2	2	1	2	0	4	0	0	0	0
3	4	2	4	2,4,3	4	0	1	4	4	4	3	5	0	5	1	0	0	0
3	5	1	5	3,4,5	0	0	0	1	0	2	3	3	0	4	0	0	3	0
3	6	2	4	4,3,2	1	0	0	3	4	1	1	0	0	5	0	0	0	0
3	7	1	3	2,3,4,1,5	1	0	2	1	2	1	0	2	0	2	1	1	1	0
3	8	1	5	4,2	4	0	0	4	2	3	1	0	0	3	1	0	0	0
3	9	1	5	4	5	0	0	5	5	5	0	0	0	5	0	0	0	0
3	10	2	5	4	2	0	0	3	3	3	2	0	0	4	0	0	0	0
3	11	2	5	4	3	0	0	4	4	4	0	5	0	5	5	0	5	0
3	12	1	5	3,4,2,1,5	3	0	3	4	3	3	3	3	0	4	0	0	2	0
3	13	1	5	5,4,3,2	3	0	0	4	2	3	0	3	0	3	0	0	0	0
3	14	2	5	4	1	0	0	5	2	3	5	5	0	5	0	0	0	0
3	15	1	5	4	1	0	0	0	0	0	0	0	0	5	0	0	0	0
3	16	2	5	3,4,2	0	0	0	1	0	0	0	1	0	2	0	0	0	0
3	17	2	4	3,2	3	0	0	3	3	3	3	0	0	4	0	0	0	0
3	18	2	5	4	0	0	0	0	0	0	0	2	0	4	0	0	0	0
3	19	2	4	2,3,4,5	4	0	0	3	4	4	4	1	0	4	3	1	1	0
3	20	1	5	4	4	0	2	4	4	3	1	0	0	4	2	0	1	0
3	21	2	4	3,4	2	0	2	3	3	3	1	4	0	4	1	2	1	1
3	22	1	4	4	0	0	0	0	0	0	0	0	0	5	0	0	0	0
3	23	1	5	4,5,2,3,1	0	4	0	0	4	0	0	0	0	5	0	0	0	0
3	24	1	5	2,4,3,5,1	3	1	1	3	1	3	1	0	0	4	0	0	0	0
3	25	2	5	4	5	0	0	3	3	3	4	0	0	5	0	0	0	0
3	26	1	5	4	2	0	0	5	0	5	3	0	0	5	0	0	0	0
3	27	2	5	3,4	5	0	0	3	3	3	3	1	0	5	0	0	0	0
3	28	1	4	4,5,3,2,1	3	1	0	3	2	1	0	3	1	4	0	2	1	0
3	29	2	5	5,4,3	5	0	0	4	4	3	0	5	0	4	1	0	0	0
3	30	1	3	4,3,2,5,1	3	0	0	2	2	2	1	0	0	2	0	0	0	0
3	31	1	5	4	3	0	0	5	3	5	5	4	0	5	0	1	0	0

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
4	1	2	2	3,4,1,2	4	0	1	5	4	3	0	3	1	2	4	4	0	0
4	2	2	0	1	0	0	0	0	4	0	0	0	0	1	0	4	0	0
4	3	1	0	3	1	0	0	3	1	3	0	2	0	0	1	1	1	0
4	4	2	4	3,2,1	5	0	1	5	5	4	4	3	0	4	3	3	3	0
4	5	1	0	3	1	0	1	2	2	0	0	1	0	0	2	1	0	0
4	6	2	0	2,3	1	0	0	3	4	1	1	0	0	1	0	1	0	0
4	7	1	0	2,3,1,4	3	0	1	2	2	2	0	2	0	0	2	2	2	0
4	8	1	1	3,2	4	0	0	1	3	0	0	0	0	0	0	1	0	0
4	9	1	0	3	4	0	0	0	4	0	0	3	0	0	0	0	0	0
4	10	2	1	1	1	0	0	2	4	1	2	0	0	4	0	0	0	0
4	11	2	0	1	4	0	0	1	4	1	0	4	0	0	4	4	2	0
4	12	1	1	3,2,1,4	3	0	2	1	1	1	0	2	0	2	0	1	1	0
4	13	1	0	2,3	2	0	0	0	3	0	0	2	0	0	0	2	0	0
4	14	2	0	3	5	0	2	3	4	3	2	5	0	2	1	5	2	0
4	15	1	0	3	5	0	0	0	1	0	0	1	0	0	0	0	0	0
4	16	2	1	3,2,4,1	1	0	1	2	3	2	2	3	0	3	1	0	0	0
4	17	2	0	3	5	0	0	1	5	0	1	3	0	0	1	1	0	0
4	18	2	0	3	2	0	1	2	3	1	0	3	1	3	1	2	0	0
4	19	2	2	2,3	4	0	0	5	4	3	3	4	0	2	2	3	0	0
4	20	1	1	3	4	0	2	1	3	0	0	3	0	0	0	3	1	0
4	21	2	1	3	1	0	1	2	3	1	1	2	0	2	3	1	3	0
4	22	1	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	23	1	1	3,2,4,1	2	2	0	1	3	1	0	0	2	0	0	2	0	0
4	24	1	0	2,3,4,1	5	0	0	3	5	0	0	5	0	0	3	0	0	0
4	25	2	1	2,1	0	0	0	4	2	4	0	0	0	2	2	4	0	0
4	26	1	0	3	4	0	0	0	1	0	0	4	0	0	0	0	0	0
4	27	2	0	2	5	0	0	5	4	5	4	5	0	3	2	0	1	0
4	28	1	1	4,3,2,1	3	1	0	1	4	1	0	4	2	0	0	0	0	0
4	29	2	0	3	5	0	0	3	3	0	0	5	0	1	4	4	0	0
4	30	1	0	3,2,4,1	3	0	0	0	2	0	0	0	0	0	0	0	0	0
4	31	1	0	3	4	0	0	2	2	2	3	3	0	1	0	1	0	0

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
5	1	2	0	1,2	3	0	1	0	5	0	0	4	0	0	3	4	1	0
5	2	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	3	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0
5	4	2	0	1	5	0	1	1	5	0	0	5	0	0	5	1	5	0
5	5	1	0	1	0	0	2	1	0	0	0	0	0	0	2	0	1	0
5	6	2	0	1	1	0	0	0	4	0	0	0	0	0	0	1	0	0
5	7	1	0	1,2	1	0	0	0	2	0	0	0	0	0	1	0	1	0
5	8	1	0	1	4	0	0	0	2	0	0	0	0	0	0	0	0	0
5	9	1	0	1,2	3	0	0	0	3	0	0	3	0	0	0	3	0	0
5	10	2	1	1,2	2	0	0	0	3	0	0	2	0	0	0	0	0	0
5	11	2	0	2	4	0	0	1	4	1	0	4	0	0	1	4	1	0
5	12	1	1	1,2	1	0	0	0	1	0	0	0	0	0	0	0	1	0
5	13	1	0	1	2	0	0	0	3	0	0	2	0	0	0	0	0	0
5	14	2	0	1	5	0	1	0	5	0	0	5	0	0	0	0	2	0
5	15	1	0	1	2	0	0	0	5	0	0	0	0	0	0	0	0	0
5	16	2	0	1	0	0	0	1	2	0	0	0	0	0	0	0	0	0
5	17	2	0	1	5	0	0	0	5	0	0	4	0	0	2	2	0	0
5	18	2	0	1,2	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	19	2	1	1	3	0	1	1	4	1	0	0	0	0	0	0	3	0
5	20	1	0	1	4	0	1	0	3	0	0	0	0	0	1	0	1	0
5	21	2	0	1	3	0	1	3	4	2	1	3	0	3	4	4	4	0
5	22	1	0	1	2	0	0	0	2	0	0	0	0	0	0	0	0	0
5	23	1	0	2,1	3	1	0	0	4	0	0	0	4	0	0	4	0	0
5	24	1	1	2,1	5	0	0	0	5	0	0	0	0	0	0	0	0	0
5	25	2	0	1	0	0	2	0	2	0	0	0	0	0	0	2	0	0
5	26	1	0	2	4	0	0	0	1	0	0	4	0	0	0	0	0	0
5	27	2	1	1,2	5	0	0	1	4	0	0	0	0	0	5	0	5	0
5	28	1	0	1,2	5	0	0	0	5	0	0	5	0	0	0	2	0	0
5	29	2	0	1	4	0	0	0	4	0	0	5	0	0	0	4	1	0
5	30	1	0	1,2	3	0	0	0	2	0	0	0	0	0	0	0	0	0
5	31	1	0	1	4	0	0	0	4	0	0	4	0	0	0	2	0	0

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
6	1	2	4	4,3,5,1,2	3	0	1	2	3	3	2	4	0	3	3	3	2	0
6	2	2	3	5,1	0	0	0	2	0	0	0	3	0	3	0	0	0	0
6	3	1	5	4,3	0	0	0	0	0	0	0	0	0	4	0	0	0	0
6	4	2	5	3,2,4,1	5	0	3	5	5	5	5	5	0	5	3	2	2	1
6	5	1	3	4	1	0	0	2	1	0	0	2	0	2	1	0	0	0
6	6	2	4	4,3	1	0	0	5	4	4	2	0	0	4	0	0	0	0
6	7	1	2	3,4,1,5	1	0	1	1	2	1	0	2	0	1	1	1	1	0
6	8	1	5	4,3	4	0	0	5	1	3	1	0	0	4	0	0	0	0
6	9	1	5	3,4	5	0	0	3	3	1	2	4	0	4	0	0	0	0
6	10	2	4	3,4	2	0	0	4	3	4	3	0	0	4	0	0	0	0
6	11	2	5	3,4	2	1	3	3	2	3	0	5	5	5	5	1	4	0
6	12	1	5	4,3,2,1,5	1	0	1	5	4	5	4	4	0	5	0	0	4	0
6	13	1	5	5,4,3	0	0	0	0	0	0	0	0	0	5	0	0	0	0
6	14	2	5	4	0	3	0	1	1	1	3	5	0	3	0	0	0	0
6	15	1	5	4,3	0	0	0	0	0	0	0	0	0	5	0	0	0	0
6	16	2	5	4,3	1	0	0	1	1	1	0	3	0	5	0	0	0	0
6	17	2	5	4,3	4	0	0	0	2	0	4	4	0	4	0	0	0	0
6	18	2	4	4	1	0	2	0	0	0	0	2	0	3	0	0	0	0
6	19	2	5	3,4,5	4	0	2	4	4	5	3	4	0	5	2	1	2	1
6	20	1	4	4	4	0	1	3	4	3	0	0	0	3	1	3	1	0
6	21	2	4	3,4	2	0	0	3	3	4	2	3	0	4	0	1	2	3
6	22	1	4	4	0	0	0	0	0	0	0	0	0	4	0	0	0	0
6	23	1	4	4,3,5,2,1	0	2	0	0	4	0	0	0	0	5	0	1	0	0
6	24	1	5	3,4,5,2,1	4	0	0	4	4	4	2	0	0	5	0	0	0	0
6	25	2	5	3,4	0	0	0	5	5	5	3	0	0	4	0	0	0	0
6	26	1	4	4	3	0	0	3	1	0	0	3	0	1	0	0	0	0
6	27	2	5	3,4	4	0	0	5	5	5	5	1	0	5	2	0	1	0
6	28	1	4	4,5,2,3,1	2	0	0	0	1	0	0	3	0	3	2	0	2	0
6	29	2	5	5,4,3,2	3	0	0	4	4	3	0	3	0	4	4	0	0	3
6	30	1	3	4,3,2,5,1	3	0	0	2	2	2	1	0	0	3	0	0	0	0
6	31	1	5	4,3	2	0	0	5	3	5	5	4	0	5	0	4	0	0

Scenario	Participant No	Version	Negligence	Evidence	BobSad	BobHappy	BobAnger	BobGuilt	BobSympathy	BobShame	BobFear	AndreSad	AndreHappy	AndreAnger	AndreGuilt	AndreSympathy	AndreShame	AndreFear
7	1	2	5	4,3,2,5,1	1	3	0	1	1	0	1	5	0	4	3	1	1	1
7	2	2	0	4,1	0	0	0	0	0	0	0	3	0	2	0	0	0	0
7	3	1	5	4	0	0	0	0	0	0	0	1	0	4	0	0	0	0
7	4	2	4	2	3	1	1	4	3	4	3	5	0	5	3	0	2	0
7	5	1	5	4,3	0	3	0	0	0	1	2	0	0	3	0	0	2	0
7	6	2	4	3,2	0	0	1	3	2	3	2	0	0	5	0	0	0	0
7	7	1	2	2,3,4,1,5	1	0	1	1	2	1	0	1	0	2	1	0	1	0
7	8	1	2	4	2	0	0	1	0	0	0	0	0	3	0	0	0	0
7	9	1	0	4	0	4	3	0	0	0	0	1	0	4	0	0	0	0
7	10	2	5	4,2,3	2	0	0	2	1	1	0	0	0	2	0	0	2	0
7	11	2	5	2,3	0	3	3	0	2	1	0	5	0	5	5	0	1	0
7	12	1	5	3,2,4,1,5	0	0	1	1	0	1	0	4	0	5	0	0	4	0
7	13	1	5	2,3,4,5	0	3	0	0	0	0	0	3	0	5	0	0	0	0
7	14	2	5	3,4	0	5	0	0	0	0	2	5	0	5	1	0	1	0
7	15	1	3	4,3,2	0	0	0	0	0	0	0	0	0	3	0	0	0	0
7	16	2	5	3,2,4	0	2	0	0	0	0	0	2	0	5	0	0	0	0
7	17	2	5	3,2,4	4	0	0	4	0	0	4	0	1	5	0	0	0	0
7	18	2	4	3	0	0	2	0	0	0	0	1	0	2	0	0	0	0
7	19	2	5	2,3,4,5	1	4	4	1	1	2	2	3	0	5	4	1	2	1
7	20	1	4	3	4	0	1	4	0	4	1	0	0	2	2	2	1	0
7	21	2	5	2,3	2	0	3	1	1	2	3	3	0	4	1	1	1	1
7	22	1	2	4	1	0	0	0	1	0	0	0	0	4	0	0	1	0
7	23	1	0	4,3,5,2,1	0	4	0	0	4	0	0	0	1	5	0	1	0	0
7	24	1	5	3,2,4,5,1	4	0	0	4	0	4	2	0	0	5	0	0	0	0
7	25	2	5	2,3,4	0	0	0	1	3	0	1	3	0	5	0	0	0	0
7	26	1	5	4	0	0	0	0	0	2	2	2	0	5	0	0	0	0
7	27	2	0	2,3,4	0	4	0	0	0	0	2	3	0	5	0	0	0	0
7	28	1	5	4,3,2,5,1	0	0	2	0	0	0	1	2	0	4	2	0	2	2
7	29	2	5	5,3,2,1	3	0	0	3	3	3	0	5	0	4	4	2	0	0
7	30	1	3	4,2,5,1	0	2	0	0	0	0	0	0	0	3	0	0	0	0
7	31	1	5	3,4	1	0	0	1	0	0	1	4	0	4	0	3	0	0