SophoLab

Experimental Computational Philosophy

Vincent Wiegel

Simon Stevin Series in the Philosophy of Technology

Stellingen behorend bij het proefschrift 'SophoLab: Experimental Computational Philosophy' van Vincent Wiegel.

1. Experimental philosophy is a useful tool for finding answers to complex philosophical questions. The very attempt to construct artificial equivalents of philosophical subject matter brings an understanding that is hard to achieve in different ways (thesis, chapter 3, 4 and 5).

2. For the set-up of a laboratory for philosophical experimentation, the philosophical community should adopt an international set of standards (thesis, chapter 2).

3. Combinations of modal logics are excellent for *modelling* required moral behaviour, and of little use in *implementing* the required moral behaviour (thesis, chapters 3, 4, and 5).

4. It is possible and desirable that artificial agents will act in a few decades as disaster relief agents. They will unavoidably have to solve moral dilemmas autonomously as they try to save human lives.

5. Further work in experimental philosophy will lend increasing credibility to a naturalist point of view on morality, at the expense of non-naturalist moral philosophies like emotivism and intuitionism.

6. The harder it proves to implement a particular moral epistemology the more likely it is that it is fruitless.

7. Software engineering should be a standard part of the philosophy curriculum.

8. Economic growth as defined in the Bruntland report and sustainability are incompatible (Verburg, R.M. and Wiegel, V. (1997) 'On the compatibility of sustainability and economic growth.', Environmental Ethics, Vol. 19, pp.247–265).

9. The adoption of lean production should be an integral part of national economic policy (Jones and Womack, Lean Thinking (1996), Lean Consumption (2005)).

10. Our attempts to construct virtuous artificial agents will create, as they begin to look more like humans, the first victims of artificial identity crisis (free after Calvin and Hobbes, 'Scientific progress goes boink', Waterson, 1991).

Deze stellingen worden verdedigbaar geacht en zijn als zodanig goedgekeurd door de promotor(en).

SophoLab
Experimental Computational Philosophy

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus Prof.dr.ir. J.T. Fokkema
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op maandag 7 mei 2007 om 12:30 uur
door Vincent WIEGEL
doctorandus in de economie
doctorandus in de filosofie
geboren te Rhenen

Dit proefschrift is goedgekeurd door de promotoren:
Prof. dr. M.J. van den Hoven
Dr.ir. J. van den Berg (toegevoegd promotor)

Samenstelling promotiecommissie
Rector Magnificus, voorzitter
Prof. dr. M.J. van den Hoven, Technische Universiteit Delft, Promotor
Dr. ir. J. van den Berg, Technische Universiteit Delft, Toegevoegd Promotor
Prof. dr. J.P.M. Groenewegen, Technische Universiteit Delft
Prof. dr. ir. P.A. Kroes, Technische Universiteit Delft
Prof. dr. J.H. Moor, Dartmouth College, Hanover, USA
Dr. L. Floridi, Universit‡ degli Studi di Bari, Italië
University of Oxford, Engeland
Dr. G.J.C. Lokhorst, Technische Universiteit Delft, Adviseur

Dr. G.J.C. Lokhorst heeft als begeleider in belangrijke mate aan de totstandkoming van het proefschrift bijgedragen.

ISBN: 978-90-5638-168-4

Voor W, M, en C, mijn medereizigers

"Elk kompas dat ik in gedachten raadpleeg, voert me naar een denkbeeldig land. Ik ben voortdurend op zoek naar nieuwe ideeën en inzichten, ik zoek geen bevestiging van wat ik al weet."

De bespiegelingen van Fra Mauro, in 'De droom van een kaartenmaker'.
Fra Mauro was een monnik die leefde in de 15e eeuw, en werkte als cartograaf aan het hof van Venetië.

"Every compass I consult in my mind, takes me to an imaginary land. I am constantly looking for new ideas and insights, I do not seek confirmation of what I already know."

Reflections of Fra Mauro, in 'A mapmaker's dream.'
Fra Mauro was a monk living in 15th century, working as a cartographer to the court of Venice.

Simon Stevin Series in the Philosophy of Technology
Editors: Peter Kroes and Anthonie Meijers

Volume 1: Marcel Scheele, The Proper Use of Artefacts: A philosophical theory of the social constitution of artefact functions
Volume 2: Anke van Gorp, Ethical issues in engineering design: Safety and Sustainability
Volume 3: Vincent Wiegel, SophoLab. Experimental Computational Philosophy

Wiegel, V.
SophoLab. Experimental Computational Philosophy
Champs Elyseesweg 24
6213 AA MAASTRICHT
E: vwiegel@xs4all.nl

ISBN: 978-90-5638-168-4

Contents - brief

Contents - detailed

Acknowledgements

One of the pleasures of writing a thesis and publishing it, is that you can share your joy with all those that helped you, and those that are dear to you. I will not list them all individually, but I do know how much I benefited from their help and support, and so do they. There a few people I would like thank explicitly.
Some of my English colleagues and acquaintances have helped me with the spelling and grammar. Writing in a language that is not one's own can be difficult. I benefited greatly from various people, native and non-native speakers alike. In particular, I would like to thank Mr. C. Burgener and Mrs. Melissa Gilmore. Also greatly appreciated in this context are the comments from reviewers of the papers and of the thesis. Any errors left are of course mine[1]. Mr. A. van Berkum, director Flex at ING, kindly provided the money to buy the software I used in this research. A brave move as we had no clue what would come out of the experiments. My dear uncle Mr. M. Wiegel put me on the programming track that got me to this stage. Mrs. Henneke Filiz-Piekhaar, thanks for your support throughout the process and the procedures. I owe a big thank you to Dr. Jan van den Berg for all the effort that he put in when he became my 'toegevoegd promotor' at a fairly late stage. The discussions were always stimulating. Dr. Gert-Jan Lokhorst has helped me throughout the whole journey. He kept me in check when the scope was in danger of expanding to undo-able dimensions. Thank you, Gert-Jan! Most of all I owe to Prof. Jeroen van den Hoven, who was involved from the start as well. When I first told him about my ideas he supported and encouraged me, and ultimately made it possible for me to write and defend the thesis at the Delft University of Technology.

All but one chapter have been published either in journals or presented at conferences. It was a great pleasure to meet many researchers, and have the ability to present ideas to them. The chapters have been modified only slightly. The introductory and conclusion chapters are new.
Chapter 2 is forthcoming in ETIN. Chapter 3 was presented at iC&P in Laval 2006 and will appear in the proceedings. Chapter 4 was presented at the CEPE 2005 conference and appeared in ETIN. Chapter 5 is currently under review. The Epilogue is based on an invited presentation that was given at the ALife X 2006 conference, and appeared in the proceedings.

[1] In this thesis I follow the British English spelling. I used the Compact Oxford English Dictionary of Current English (OED), third edition. It favours the use of the –ize suffix over the -ise in many cases: e.g. organize rather than organise. I left the spelling in quotes as it is in the original text.

1 Introduction

This thesis is about empirical research in moral philosophy and computational experiments. The availability, or lack thereof, of empirical data to validate moral theories has been an important issue in moral philosophy over the last decades. The point was poignantly made by Quine (1981) in his assertion that ethics is 'methodologically infirm' because it lacks responsiveness to observation. Whether Quine's observation upholds is a matter that will not be addressed in this thesis. I do acknowledge though, that in moral philosophy the empirical data and testability of moral theories are issues that are worth attention. Turning our attention to these aspects will help providing additional means to better understanding of, create enhanced insights into the implications of, and provide support for moral theories

In the history of moral philosophy the empirical question has long not been on the foreground. Starting with meta-ethics moral philosophy focused on the questions about the nature and source of goodness, what it means to say that something is right or wrong, whether we can know what good is, and if so how. In normative ethics these preceding questions are (implicitly) assumed answerable, and mostly affirmative. Normative ethical theories defend particular positions or norms as morally good and refute others. It is concerned with the development of these norms and the classification of actions and intentions according to these norms. Applied ethics studies these norms in practical settings and dilemmas. Starting from a particular normative position it investigates the application to specific cases. Applied ethics has all kinds of sub-branches based on the specific application domains such as medical ethics, environmental ethics and business ethics. This applied 'turn', as one could call it, takes ethics away from its predominantly theoretical province to the world of everyday lives. Of course, in both meta-ethics and normative ethics there is reference to practical situations. This reference, however, is not systematic and often suffers from artificiality. The application of ethics to practical situations led to powerful methods such as the wide reflective equilibrium in which we adjust various normative positions in view of practical applications such that judgment is reached about what normative positions to accept. The latest branch in moral philosophy is empirical ethics. Where applied ethics is primarily concerned with bringing ethics to the realm of everyday life, empirical ethics seeks for systematic ways to find empirical proof and refutation for normative positions. In its aims it goes to the root of the issue of the methodological infirmness of ethics. The focus in this thesis is on empirical ethics.

In empirical ethics various methods are used. Working at the intersection of economics and ethics Frank (2005), for example, uses game theoretic experiments to investigate the emergence of non self-interested behaviour patterns. One of the experiments involved economists and non-economists. With this experiments Frank could establish to what extend knowledge of the neo-classical model of narrow, self-interested rationality does effect the behaviour. And indeed, marked differences between the economists and the non-economists emerged from the experiments. The former displayed a clear tendency to free-ride whereas the latter displayed behaviour that was more tuned towards lasting relationships that could not be explained by the traditional narrow understanding of self-interest.

Doris and Stich (2005) deploy a method that is "a resolutely naturalistic approach to ethical theory squarely engaging the relevant biological, behavioural, and social sciences". The main contention underlying their approach is that the psychology and anthropology underlying moral theories should not be 'invented' by moral philosophers, but based on empirical data gathered by other sciences. This data is not easy to come by as they note, but can be of great value to the moral debates. A striking example is in virtue ethics, 'what sort of person to be?', where the focus is on character traits, and their associated emotional and cognitive patterns. The traditional conception of character, assuming that "traits with associated evaluative valences are expected to co-occur in personality" (Stich, 2005:118), underlying virtue ethics is deemed inadequate based on empirical, psychological studies. These studies showed that virtuous behaviour is strongly dependent on the situation in which the behaviour is displayed. For example, "people who had just found a dime were twenty-two times more likely to help a woman who had dropped some papers than those who did not find a dime". Another example cites levels of cortisol and testosterone[1] in situations of offence, hurt pride and honour. These are notions that are related to morally relevant character traits and behaviour, and turn out to have a strong biological component. And more to the point, these biological traits were closely associated to different cultures of honour. Whatever the resulting moral position, it shows that empirical data have a contribution to make to the moral philosophical debates. As 'ought' implies 'can', the above biological data could possibly show that what we consider appropriate moral behaviour is in fact biologically impossible, or unlikely to happen. Thus it questions the force of the 'ought'.

Yet another method uses the computer to generate empirical support for theories. In this method the philosopher constructs models as approximations of

[1] These are hormones associated respectively with high levels of stress, anxiety and arousal, and with aggression and dominance behaviour.

reality. These models are implemented on computers. Using the computer to create executable models is not unlike the work in laboratories. The computational method constitutes an approximation of empirical data. The model, based on a theory, generates information that provides both support for the theory, and deeper insight into the implications of that theory, in a way that cannot be said to be analytic or synthetic. As Bynum and Moor (2002: 3) state it "Philosophy done with computers has an empirical dimension that distinguishes it from philosophy typically thought of as conceptual analysis or synthetic construction." Though the computer model obviously is not a human moral agent it is a useful substitute, in a similar way as a rat is obviously not a human patient but still provides useful insight into the functioning of biochemical processes. The laboratory setting allows for a higher degree of control and abstraction. The control is achieved because the experimenter calls the pace, and is able to configure the experiment. All non-relevant aspects are abstracted from which makes it a particularly well suited method for complex situations. Danielson (1992) offers a paradigmatic example of the computational strand in empirical ethics, and is intensively analysed in this thesis. He constructs software agents whose behaviour and strategies 'embody' particular moral positions. These agents are then matched in a game-theoretic setting. This allows Danielson to investigate both the understanding, and the support for strategies and moral positions proposed. As particular positions proved to be procedurally impossible this highlighted the original lack of understanding of the theory, and/or unclarity about what the theory implies.

The research in this thesis sits predominantly within the empirical tradition. It seeks to develop an experimental[2] approach within the computational method. The computational method refers to the method in which computers or computational processes are used as method and as model for philosophising. As Bynum and Moor (2002: 2) express it: "Computer models are extremely fruitful for philosophical reflection. Because computational processes are so logically malleable, they provide intellectual clay that can be shaped to formulate ideas, explain events, and test hypotheses." It is in this, instrumental, sense that in this thesis theories are justified or find empirical support. A theory that is found procedurally impossible cannot be a proper theory; a theory that makes claims that cannot be produced through experiments is failing a test.

[2] The experiments in this thesis and the experiments that are referred to are for the largest part computer-based experiments. These could be named simulations as well avoiding the sometimes connotation of reference to a 'real' object that is used in the experiments. I will stick to the terminology of experiments because it is used as such in the literature referred to, and it emphasis that it is an extension the traditional philosophical method of thought experiments. Key in the understanding of the notion experiment is the controlled test of a theory under laboratory conditions.

'Empirical proof', 'testability' are terms that might raise some eye brows. The danger of committing the naturalistic fallacy, if indeed a fallacy it is, is nearby[3]. G.E. Moore (1903) argued that moral properties cannot be inferred from, or defined by reference to natural properties. There are three aspects to experimentation that are relevant in this context. One, gaining better understanding of the theory at hand, in particular aspects as hidden assumptions and dependencies; two, the creation of enhanced insights of the implications of a theory; three, support for, or refutation of a theory based experimental data gather.
It could be argued that, since experimental, computational philosophy is about tools and techniques for experiments, and not about constructing moral theories, the naturalistic fallacy cannot be committed simply because no theorizing is done. This would, however, only delay the objection because it would confront the philosopher using experiments in his theorizing. Aspects one and two can, even so, be said to be about uncovering what is in the theory already. And thus, not be open to the objection. Aspect three of support and refutation, the challenge might be harder to meet. If one were to deduce that "since the experiments show such and so to be the case, X must be good". This would be a rather simplistic use of experiments. Empirical support in this context means that the experiments provides means to evaluate (contrary) claims. E.g. if it is argued that rule utilitarianism is to be preferred over act utilitarianism because it generates more utility, this claim can be investigated. Data shows that one or the other generates more utility, and hence gets the empirical support. Empirical support does not mean that we can have empirical support for the moral claim that more utility is morally preferable to less utility. Understood in this sense, experiments cannot be claimed to be open to challenge of the naturalistic fallacy. The underlying theory is, of course, still open to that objection[4]. This thesis contains a methodological chapter, chapter two, focusing on what it means to use computers to create test environments for normative theories. It is based on various research done in the field of computational philosophy and in this thesis. It contains also a study of a case from philosophy of science (Thagard) to illustrate the likeness in approach. Two chapters contain discussions on experiments conducted as part of this research. One experiment (chapter three) is set up to gain a better understanding of the nature of negative moral commands ("thou shall not..."). In the process of constructing artificial agents with an ability to use negative

[3] I am particularly grateful to Luciano Floridi for his remarks on this issue.

[4] Yet another line of argumentation is to address the naturalistic fallacy argument itself. Various authors have argued that the naturalistic fallacy, understood as non-naturalism and non-reductionism, is both not a fallacy (Frankena, 1939, Williams, 1985) and various counter arguments. Engaging in this discussion is clearly outside the scope of this document.

moral commands it becomes clear that our understanding of negative moral commands might require revision.
Another chapter contains a discussion that fits better under the header of applied ethics (chapter four). It stands apart from the other chapters as it looks at the application of a theory, rather than trying further understand and support that theory. It uses the tools developed for experimentation to suggest practical solutions to moral challenges. It is motivated by the consideration that "The task of developing artificial moral agents becomes particularly important as computers are being designed to perform with greater and greater autonomy..." (Allen et. al. 2005:21) . It shows that the tools and techniques used to investigate and empirically support normative positions, can also be used to apply these normative positions to the very same practical situations. And of course, in the end the proof of the pudding is in the eating, or in this case, that the proof of theory is in the application.

A major driver for the development of the approach and experiments in this thesis is complexity. The subject matter of normative ethics is a world of a great many relationships. In order to derive empirical support for normative positions, practical cases with their inherent complexity must be investigated. This is hard to do without computerized support. From the perspective of application it is the same complexity that drives the need for computerized support. The consistent application of particular norms throughout a complex situation is not possible without computerized support either.

1.1 Daunting complexity

Imagine the buy of a new house and taking out a mortgage to finance it. The mortgage is linked with a life insurance to ensure that in the event of your death the loan will be paid, and your dear ones will at least not have the financial worry added to their grieve. How many people, do you think, are involved in processing your application?
You discuss your financing needs with your financial advisor (1), assessing your financial situation, checking with your account manager (2) at the bank where you keep your assets, getting a statement from you employer (3) stating your income and nature of your contract (permanent or contractor). Your adviser helps you filling out all the necessary forms, which are completed, stamped and sent by the secretarial support staff (4) of your financial advisor. The life insurance will be assigned to the bank that provides the loan. The application form is sent to the insurer where the office support staff (5) scans the correspondence, except for the medical information, which has been provided in a separate envelope. The administrative staff (6) receives notification of the application and

processes the form. The medical administrative staff (7) pre-processes the medical information. In case there is something potentially deviant from the standard health conditions (about weight, medical history, parents' health) the application is forwarded to the medical assistant (8) who assesses the medical condition. He orders some extra blood tests to verify the potential health risk and inquires with your general practitioner (only after you have consented). The request is sent to you and your general practitioner, the blood tests are run by the laboratory personnel (9), and all results are forwarded to the insurer. Nothing very serious, but a small increase in premium is required to cover for some additional health risks. This requires the formal approval of the insurer physician (10). The factoring company / companies (11) generate(s) invoices for your visits to the GP, and the laboratory. The necessity of the additional premium is discussed with the administrative staff and the team leader (12) because you are a valued customer. The insurer administrative staff produces your policy document, and forwards it to the bank to which it is assigned. The bank's office support staff (13) records the receipt of the policy. In the mean time, since the whole process is taking a long time and the date at which the sale of your house is to be finalized is coming close, your financial adviser has called the mortgage provider. At the help desk (14) staff could not find the status information of the insurance, and forwarded the call to the second line support (15). Second line support sees you have applied for a life insurance at insurer *X* and calls their help desk (16), which looks up the status information at their system, and return the information, etc. The bank administrative staff (17) processes the loan application and prepares all documents, notifies the payment department (18) that the money can be paid to the notary (19) who conducts the transfer of the property. The notary assistant (20) prepares all necessary contracts, deeds, etc. After all documents have been signed the money is transferred from the notary's bank account to the bank account of the seller.

This example of a transaction that most people will conduct a few times in their life is by no means exceptional. Even in this description the number of actual people involved will be larger, think of illnesses and colleagues taking over, more complicated tests, file transfers to other teams, quotations, checks on credit worthiness, marketers requiring information of target groups, etc. The information involved consists of at least four subsets of information: personal information on age, marital status, etc.; loan related information like type of loan, amount, etc.; life insurance related information like type of insurance, life assured, etc.; medical information of the insured person. All these sets of information and their processing are guided by particular legal rules, company internal codes of conduct, and moral rules. The complexity of such a situation is

daunting indeed, and poses particular challenges to our thinking about moral (and other) rules.

Complexity as described here challenges both the development of theories as well as the application of these theories. How does a moral philosopher come to a full understanding of the implications of his theory in such complex situations? And next, once properly understood, how is the application arranged in such a way that is both doable for the people involved, and correct and in line with the theory? The privacy issue above is a point in case. Defining what is right and wrong is hard when overseeing all interactions and consequences, if not downright impossible, in the traditional way. Once defined, however, the application is far from trivial either. How is one to safeguard the privacy in situations as described above with all these different roles, interactions and transactions?

There are three drivers for this thesis. The first, already alluded to in the introduction, concerns the improvement of theories through better understanding, a greater ability to assess the implications of a theory and the ability to test the theory. In implementing moral theories and rules, the nature of these rules, of moral reasoning, etc. is better understood. The attempt to implement will show some of the blank spots we still have in our understanding of morality and moral reasoning. Implementation in computer systems requires an exact and complete instruction consisting of conceptualizations and strict reasoning, because the computer by its very nature has no knowledge of its subject-matter. Although a very obvious statement, it is has far reaching consequences, because most moral theories and their application assume shared understanding of the subject matter by the experts, and on much implicit knowledge, that might prove hard to explicate.

The second driver is closely related to the first one. In these complex situations the investigation, correct design and implementation of moral rules can no longer be done without the aid of sophisticated tools. The complexity is too large to oversee, and in the interaction between the various people involved new characteristics can arise as emergent trait of the system. To provide these tools moral philosophers (as well as legal researchers and software engineers) need to create an artificial environment in which they can construct these situations and investigate the properties of the system, the consequences of their decisions and the unforeseen traits of the complex interactions.

The third driver is that, as technology advances, artefacts start to play increasingly important roles in our lives. They do not only contain information about us, they can also, and in fact often already do, act on our behalves. With their increasing autonomy comes an increased need to ensure that their behaviour is in compliance with what we expect from them. We want to ensure that no abuse is made when an opportunity offers, no harm is done to others, etc. Besides this

negative, constraining aspect there is also a positive aspect. If we want to develop machines that can help us, for example, in circumstances that are too dangerous for humans to intervene, they will need all the reasoning power we can give them including moral reasoning. To illustrate this point imagine the following. With the advancement of robotics it is conceivable that artificial relief agents will be deployed where human agents finds it hard to come and operate, e.g. flooded areas, mountainous villages far from roads hit by an earthquake. As these artificial relief agents will also be faced with limited resources they might have to make morally loaded decisions about the distribution of aid.
In all the above claims it remains to be seen whether the computer models are powerful enough to capture the complexity in full. In chapter 3, I will argue with Moor and Wooldridge, that they are by far not powerful enough to capture reality as we know it in everyday life. The underlying claim, this thesis sets out to demonstrate, is that they are powerful enough to capture philosophical theories. As far as the use of computers and computer models is concerned the claim is neither that they are the only possible tools, nor that they are the best suited tools. They are subject of this thesis because over the last decade they have emerged as the most often used tools by philosophers. In the end, the hope is to make a contribution to the 'operationalization' of artificial morality. As (Allen et. al. 2005) put it: "The development of an effective foundation for the field of artificial morality involves exploring the technological and philosophical issues involved in making computers into explicit moral reasoners."

This introductory chapter aims to delineate the research area and to provide the background to the various themes and developments in the field of formalization of moral philosophy broadly conceived. This chapter is organized as follows. The above drivers places this research at the intersections of various research areas. These areas are described in section 1.2. This helps to delineate the research of this thesis. To further clarify and restrict the research domain some recent developments in these fields of research are discussed in section 1.3. In section 1.4 I provide an introduction on how the idea for this research arose. The hypotheses and detailed research questions are presented as well as an overview of the thesis lay-out.

1.2 Combining four research areas

This research project touches on four major research areas. The type of experimentation I envisage requires an application domain, and tools and techniques to conduct the experiments. Thus, by its very nature, the research must be multidisciplinary.

In this research the subject domain is moral philosophy (1). The research questions are cast in modal logic (2). The experiments make use of software agents (multi-agents systems) (3). And, finally, the approach stands in a tradition that has become known as computational philosophy (4). Each of these is discussed in the following sub-sections.
The multi-disciplinary nature is both an enriching and a complicating factor. On the interfaces between different research areas interesting cross-fertilization can take place. The application of an idea from one area to another can shed new light on an old problem. The research benefits from the combination of intellectual efforts from different areas with different modes of thinking and casting of problems. On the other hand, the multi-disciplinary approach does require a broad overview across the various research areas. An in-depth overview is out of scope. The inherent risk is, of course, that one omits important and relevant work that has been done. This risk will have materialized, known (in retrospect) and unknown. Still I feel the benefits outweigh the risk.

1.2.1 Moral philosophy

The approach of setting up experiments to investigate various, at times opposing, positions intends to be neutral in the sense of not taking sides with any of the positions. E.g. later in this thesis a discussion on rule and act utilitarianism is subjected to experimentation. These two strands of utilitarianism have opposing claims as to which version is best for welfare maximization. In setting up the experiment there is no idea a priori as to which one is to be favoured. Nonetheless, it seems almost inevitable that some positions will be taken. This will arise due to restrictions of my approach, the choice of modelling tools and the scope. Though I, obviously, cannot list them all, a few examples are now discussed.
Taking the Bratman's Belief-Desire-Intention model (BDI-model, Bratman, 1987) as a starting point various (meta-)ethical positions will follow. The approach will be closer to the internalist position, allowing a relation between believing something to be good and the formation of an associated intention to do the good, than to the externalist position. According to the externalist position there is no internal connection between the relevant cognitive and motivational states. Another example is using logic as the primary language to specify moral characteristics of agents. This has the implication that the characteristics will have a tendency to be thin rather than thick concepts[5]. This tendency is further

[5] This distinction between thin and thick moral concepts was introduced by Bernard Williams (1985). Concepts or terms that leave open the exact nature of what constitutes them are called thin concepts. What good means, or what it is, is not clearly defined. The more content is added to a concept the thicker it becomes. Loyalty, altruism, etc. are good moral traits. We can still debate about whether they are required morally, and

strengthened by the unavoidable limitations in scope. Thicker moral concepts, like for example loyalty, can probably be expressed in modal logic terms. In this thesis the focus is still on the more basic, and 'thinner' concepts that need modelling, before one can move to the next stage of 'thicker' concepts.

In any implementation of (artificial) moral reasoning the notion of obligation is at the very heart. At the level of the meta-ethical discourse various answers have been given to the question 'What are moral obligations?' and 'Where do they originate from?'. The intuitionist states that moral truths are irreducible properties of which we have an intuitive awareness. To the prescriptivist it is all about making prescriptions and recommendations for actions and decisions. For the emotivist statements about obligations express a feeling we have towards a course of action. And for the descriptivists, it is referring to an attribute that is essentially connected with certain natural properties of objects.

When implementing moral reasoning in artificial agents the above questions do not have to be answered, nor is there a need to provide an account of the answers provided by various moral philosophers. I am looking at the meta-level at the structure of the answers and concepts used in both normative ethics and meta-ethics discourse. Using this analysis I hope to provide moral reasoning capabilities that can be tuned or configured according to the moral persuasion we want the artificial agents to be equipped with. This is the creation of the laboratory 'equipment', the analysis and creation of the instruments and tool necessary to construct the experiments.

An important distinction in the moral discourse is between teleological and deontological approaches. The former grounds the good or right of an act in the outcome it achieves. The latter attributes the qualifications 'good' or 'right' to an action according to some intrinsic value of that action ('telling the truth is good no matter the outcome')[6]. The key characteristic, from the meta-level, modelling perspective, is the assigning of a value to a moral attribute of either an action or a situation. Here again, the intention is to not take any stance.

It is important to note that moral philosophers and moralists tend to base the judgement on what a moral agent could reasonably be expected to foresee as outcome of its actions. This is dependent on the knowledge and information the agent possessed, or could reasonably be expected to have, or have been able to acquire at the time.

---

what exactly counts as altruistic behaviour. But everyone will agree they refer to something more concrete than the term good. Thinner concepts are, we might say, supersets of thicker notions.

[6] This is of course a gross oversimplification of the deontological and teleological positions, and serves only to draw attention to its basic structure.

An informal characterization of the moral discourse would look like this.

1) There are actions, situations and/or persons, that can be said to be good, right, wrong, virtuous, etc., and ought to be done, avoided, brought about, etc. on moral grounds
2) There are actions to assign a moral predicate to actions, situation and/or objects, and an ability to act in consequence of this assignment
3) The actions, situations and/or persons with their predicates can be ranked either ordinally or cardinally. Particular actions, situation and/or objects can be said to take precedence over other actions, situation and/or objects. Reasoning about various actions, situation and/or objects is done at the meta-level.
4) Additional assumptions made, supporting the above, are the ability to hold beliefs; to form intentions to act; to act and interact with other agents; to be able to reason and engage in meta-level reasoning; to be logically consistent.

This very general and informal description should serve as background for the discussions in the various chapters. A requirement to the logic and software used in this research is that they can capture all concepts from this description. Which is one of the goals described in section 1.1.

1.2.3 Modal logic

Modal logic deals with modalities, expressions that state or predicate something other than the assertion of a fact. In its basic form "Modal logic is the logic of necessity and possibility, of 'must be' and 'may be'." (Hughes and Cresswell, 1996,ix). Over time other modalities, such as ones dealing with knowledge (epistemic modalities dealing with 'knowing' and 'believing') have been added to the basic alethic modalities of 'necessity' and 'possibility'[7].

When talking about agents as I later will, that is as entities that have intentions, the use of modal logic is a natural choice. Various branches of modal logic have been geared towards notions such as beliefs, intentions and obligations which play a pivotal role in this thesis. Using modal logic moral propositions can be formalised. The benefit is twofold. First, it brings rigour to otherwise loosely defined concepts and relationships. Second, it makes it easier to implement in software because the formalisms of modal logic are closer to the formalisms of programming languages than natural languages are.

Modal logic has more to offer than classical first-order logic. There are several reasons for favouring modal logic over first-order logic. First, expressing propositions expressed in first-order logic will be much harder to read, and more

[7] For an introduction in modal logic the reader is referred to (Hughes and Cresswell, 1996).

cumbersome to construct, than the same propositions expressed in modal logic. Second, classic propositional calculus is truth functional whereas intensional notions such as beliefs are not. My belief that lying is forbidden is not exclusively dependent on lying being forbidden. Of course, 'I believe that lying is forbidden' is also a proposition with a truth value. Replacing lying by an equivalent expression does not necessarily preserve the meaning of the proposition. The referential opaqueness of intensional notions means that the substitution rules of the classical proposition logic do not apply. Hence first-order logic would be more restrictive in its use (Wooldridge, 2000, 181). Thirdly, one easily runs into syntactical problems when trying to construct propositions about modal notions in first-order logic.

Using modal logic is, however, not without disadvantages. One is confronted with various decisions. Which system does one use S4, S5,...? What stance to take on non-monotony? Can the system be axiomatised? And so forth. These are all important issues, and could take the best part of this entire thesis. They are not key to my research aim which is to show, if and how, one can model and implement moral theories in order to investigate them. To avoid these issues modal logic will be used in a non-axiomatic, informal way. The use is geared in a different direction.

In software engineering logic can play three different roles (Wooldridge, 2000, 163; Halpern, 2000)[8]:

- as specification language
- as implementation language
- as validation language

In this research modal logic will primarily be used as specification language. When it comes to implementation there are various options (Wooldridge, 2000, 165): direct execution, compilation and manual translation. Key is the environment in which it needs to be executed, the JACK agent language (Agent Oriented Software). To date there exists no modal logic compiler, nor does JACK offer functionality to directly write modal logic propositions in its coding tools. Hence, in this thesis the propositions are manually refined and translated to software code.

There is another reason to stick with the informal use of modal logic. Research by Halpern (1992) indicates that satisfiability and validity for modal logic systems are PSPACE complete[9]. PSPACE complete decision problems are a class of

[8] G-J. Lokhorst pointed this out to me.

[9] This goes for K, T and S4 systems with with one or more agent(s) and for S5 and weak S5 systems with more than one agent (Halpern and Moses, 1992).

problems that is suspected to be outside of the classes of decision problems that can be solved in polynomial time on a (non)deterministic Turing machine. Practically speaking this means that automation might not be a feasible option (Wooldridge, 2000). By giving up some of the rigour the practicality can be saved.

Finally, humans get by alright in solving all kinds of problems and dilemmas (though obviously there is lot left to be improved) without the use of formal logic. Given the drawbacks of the formal use of modal logic as outlined above together with the ability to be successful without it, it seems proper to use modal logic to reason about artificial agents rather than have them reason using modal logic in an axiomatized way. (Halpern, 2000) put this argument eloquently in his discussions with McCarthy.

> "McCarthy wants robots to reason with a formal logic. I'm not sure that is either necessary or desirable. Humans certainly don't do much reasoning in a formal logic to help them in deciding how to act; it's not clear to me that robots need to either. Robots should use whatever formalisms help them deciding. Robots should use whatever formalisms help them make decisions. I do think logic is a useful tool for clarifying subtleties and for systems designers to reason about systems (e.g., for robot designers to reason about robots), but it's not clear to me that it's as useful for the robots themselves."

Others have followed this line of argumentation. Modal logic is applied and used extensively for agent modelling and systems design (Broersen et. al., 2001a; Rao and Georgeff, 1992; Wooldridge, 2000; Sergot and Richards, 2001). I feel comfortable that the informal use for specification purposes is acceptable in use for experimental, computational philosophy.

Other researchers are using defeasible logic as their main modelling tool, notably Dumas (2002), whose work is mentioned in section 1.3.3. The use of defeasible logic seems to be not as wide-spread as modal logic. The choice for modal logic over defeasible logic is motivated by the following considerations. One, modal logic has a longer, and more extensive, history than defeasible logic. This shows in, for example, the availability of decision procedures. Britz (2006:8)

> "Non-monotonic logics [...] do not have a well developed theory of effective decision procedures at their disposal [...] In contrast, there are many effective proof (and decision) procedures for modal logics resulting from an increased awareness of the range and significance of computational applications of modal logics..."

Two, there is a good fit between the BDI-model and the various modal logics. Three, for modal logics as required by the BDI-model industry strength software is available, which is not (yet) the case for modal logic. Thus, it would require more programming effort that reach the same level functionality. Four, limitations of time and resources did not allow that adoption of both. The choice for modal is empathically not a choice against defeasible logic.

To model agent behaviour the belief-desire-intention model, BDI-model (Bratman, 1987) provides a good foundation. Besides the modalities of belief, desire and intention, it captures both bounded rationality, and the goal oriented aspect that is required for autonomous agents. These are all important elements in pursuing the goals of equipping artificial agents with moral reasoning capacity that can be used for experimental purposes. There are two important elements missing in the BDI-model to make it suitable for modelling artificial moral agents: the deontic element and the action element. Therefore the BDI-model is extended through the deontic-epistemic-action logic (DEAL) framework (Van den Hoven and Lokhorst, 2002). The deontic logic covers the deontic concepts of 'obligation', 'permission', and 'forbidden'. Epistemic logic expresses the things we know and believe. And the action logic element through the introduction of the STIT - see to it that ñ operator covers the reasoning about actions. A more detailed discussion can be found in chapters three and four.

1.2.3 Agents, agents and agents

There are many labels for agents: software agents, intelligent agents, rational agents, artificial agents, moral agents, negotiating agents, etc. In addition, there is artificial intelligence, multi-agent systems, computational societies, etc. In short, there is a wide range of non-standardized terms that are closely related, and often overlap. For the purpose of this thesis it is desirable that the reader knows which terms will be used, and what their intended meaning is. In reference to various sources I will sketch an overview of these terms, and indicate which one will be used. The use of particular definitions is not to imply that I take a stance in the various discussions in which they are put forward. I merely find them useful for the purpose of this research.
In this thesis the reader will find the following terms: 'artificial agent' and 'multi-agent system'.

(Jennings, 2000, 280) defines agents as follows

> "An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives."

This definition is relative meagre compared to other definitions still to follow, but it provides a working definition. It emphasizes the situatedness of the system, and the autonomy of the system. Jennings' definition is also tied to computers only. Still, it is a wider notion, as it applies beyond PCs. On the other hand, it is restrictive as it excludes, for example, robots as computer with additional hardware. (Wooldridge 2000, 2002) definition is more encompassing and richer. He defines rational (software) agents by the following characteristics.

1) Situated in an environment.
2) Autonomous; meaning that the agent has its own desires, intentions and beliefs.
3) Pro-active; goal directed, exploiting serendipity.
4) Reactive; displaying stimulus-response behaviour.
5) Social ability; ability to interact with other, self-interested agents through cooperation and negotiation.

Wooldridge's model captures the core of the term in the sense in which I will also be using the notion. The adjective rational refers to the ability to choose that course of action that best serves the agent's goals given its beliefs, and limited processing capacity.

Artificial agents are agents in the above sense, but stressing the non-human or non-biological aspect. In this sense it is a broader concept than 'software agents', which is restricted to the computer implementation as meant by Jennings. This is the sense in which the term will be used in this thesis[10]. A multi-agent system will be the collection of artificial agents that have the ability and capability to interact.

Note that the adjective 'artificial' differs from the use in 'artificial life sciences' which are the "...emulation, simulation and construction of living systems." (Adami, 1998, 2) . Since this can also include computer-based simulations that represent living mechanisms it concerns clearly a wider meaning. Much depends also on the definition of life that is used. Recent developments in artificial life make its field of research even wider and inclusive of software agents. The core of the various definitions of life have developed from physiological to metabolic, to biochemical, to genetic and finally to thermodynamic (Adami, 1998, 6). In the latter definition:

[10] When it is obvious from the context what is meant I will use the term 'agent' for short.

> "Life is a property of an ensemble of units that share information coded in a physical substrate and which, in the presence of noise, manages to keep its entropy significantly lower than the maximal entropy of the ensemble, on time-scales exceeding the 'natural' time-scale of decay of the (information-bearing) substrate by many orders of magnitude."

Though interesting and helpful in understanding the wider setting of agent research I will not rely on this notion artificial life.

The next domain to touch upon is artificial intelligence. Again, a tremendously wide research area with many varying opinions on what it is about. McCarthy already lists eleven branches and indicates that certainly some will be missing. When the term was coined it referred to a study that would "..proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it."
According to (Marr, 1990, 133) it is "...the study of complex information processing problems that often have their roots in some aspect of biological of information processing."
(Boden, 1990, 1) provides a few suggestions (not necessarily her own opinions) that are wider than Marr's.

> "...the science of intelligence in general (with) the goal to provide a systematic theory that can explain (...) both the general categories of intentionality and the diverse psychological capacities grounded in them."

This has a clear human aspect to it. The other suggestion is geared more towards the artificial implementation aspect in which AI is "...the study of how to build / program computers to enable them to do the things that minds can do." This latter notion is helpful with respect to the intention of this thesis to show that implementation of moral reasoning in multi-agent systems is possible. The moral reasoning part reflects something that 'minds can do' and the multi-agent systems are computer programs. And thus this research fits within the definition of AI. Important is to note that reference is made to 'things that mind can do' which is a phrase that does not in anyway requires that the computer programs are (functioning like) brains. The achievements are comparable, but not their substance. Though many elements used in the tools and techniques come from AI I will not make further reference to AI, or the 'intelligent' to avoid the many possible connotations that become attached over the years.

1.2.4 Computational philosophy

The complexity in situations as described at the beginning of the chapter makes it hard to predict how a philosophical theory will behave if applied in practice. In order to investigate such situations testing in a controlled environment is desirable. This will not only help us to understand the intricate dynamics that arise in such complex situations, it also helps to validate the theory in question with respect to completeness and consistency. For this purpose the computer provides an ideal tool. Before an implementation in a computer environment is possible the situation, both theory and its application, needs to be modelled such that it can be implemented at all. As the computer is a dumb machine that needs detailed instructions it requires concise and detailed modelling.

This kind of enterprise is within the realm of what is now called computational philosophy. The introduction of computing has changed and is changing the way some philosophers view their profession, conduct their research and interpret their subject matter. As (Bynum and Moor, 1998,1) put it:

> "Computing provides philosophy with such a set of simple, but incredibly fertile, notions ñ new and evolving subject matters, methods and models for philosophical enquiry. Computing brings new opportunities and challenges to traditional philosophical activities. As a result, computing is changing the professional activities of philosophers, including how they do research, how they cooperate with each other, and how they teach their courses. Most importantly, computing is changing the way philosophers understand foundational concepts, such as mind, consciousness, experience, reasoning, knowledge, truth, ethics and creativity."

The research in this thesis is very much part of this new approach. It is this area where this research is hoped to contribute most. In particular, on the following four topics:

1) methodology of computational philosophy,
2) the understanding of moral commands,
3) the role of information in the act vs. rule utilitarianism debate, and
4) the conceptualization of deontic constraints in information relationships.

Ad 1) The various philosophers working in this new field are themselves subject matter in the attempt to identify and define a methodological framework underlying their research. Using the computer as a tool to set up new methods of research requires a methodology that acknowledges the core fact that philosophical theories are translated from one 'language' (of philosophical tradition) to an entirely different one (of computer code). Another important aspect is that of

transparency in experimentation. With the use of a, for philosophers 'new', language of computer coding a new source of arguments arises. As this source is less well known and harder to penetrate than the traditional purely paper-based argumentation, transparency must be secured. A methodological framework should provide guidelines on how this can be achieved.

Ad 2) As active practitioner of computational philosophy I set up experiments to test philosophical theories. One of these test concerns negative moral commands such as, "thou shall not...". Setting up the experiments and implementing the moral propositions results in a different understanding of negative moral commands. Negative moral commands do not refer to actions but rather to classes of actions that are identified by their effects. This insight arises as one attempts to implement an agent that knows how to not do something. And as this proves problematic, and different venues are investigated, new insights arises. Thus I demonstrate that as the experiment can be fruitfully conducted the research goals of this thesis can be achieved.

Ad 3). There is a long standing debate on the merits of act and rule utilitarianism. These two strands of utilitarianism share most of their fundamentals. They differ in some important respects however: one, does an agent need to take into account the behaviour of his fellows as an exogenous or an endogenous variable; two, is the decision to be based on the basis of the individual situation, or on the generalized rule that is applicable. Using computer models of utilitarian theories helps shedding new light on this long standing debate. By studying one of the advocates of rule utilitarianism, Harsanyi, and implementing his version of utilitarianism, I demonstrate the plausibility and usefulness of the experimental approach. Amongst other things it becomes clear that: a) the representation of the theory is underspecified; and b) the conclusions drawn cannot be generalised.

Ad 4) The last topic proposes a reconceptualization of information as an intentional agent. Whereas information is usually seen as a passive entity, which it has been so far, the experimental efforts have led to this new insight that for implementation purposes information can be fruitfully treated as an active agent. The informational relationship subject to experimentation is that between a person as patient and economic agent on the one hand, and physician and interested parties on the other hand that process medical information about this person. This relationship is deontically constrained, that is, there exists a web of obligations, permissions and rights regarding the medical information. E.g. particular information cannot be shared without the patient's prior consent, other information ought to be shared, etc. The web of relationships that is concerned is complex and extensive. Maintaining the moral integrity of the relationship and honouring the deontic constraints, is very hard to achieve

without the support of computerized tools. The approach proposed here is to encapsulate the information by an artificial agent whose goal it is to maintain the integrity of the data. This approach nicely illustrates how moral considerations can be implemented by using agent software. It also demonstrates how agents can be constructed which potentially act semi-autonomously on our behalf. This ties in to the third claim underlying this research: the increased autonomy of artificial construct that act on our behalf of which we want to be sure that they will act in a morally appropriate way. What better way to illustrate this than by designing agents that actively seek to protect our interests and serve our moral goals?

1.3 Recent research themes

In order to further clarify and delineate my research area this section reviews some recent research. When discussing artificial agents with moral reasoning capabilities, what it is exactly that is referred to? In section 1.3.1, I discuss a classifications, and specify what definition shall be used in the context of this thesis. When constructing artificial agents the question arises to what extend these can be held morally responsible for their actions. This discussion touched upon in section 1.3.2. In section 1.3.3 the moral theoretical aspects of the approach taken are discussed. And the decision to use the belief-desire-intention model is argued for. The approach is contrasted with some other notions that are used to characterize moral theoretical positions. There is a lot of research underway concerning constructing artificial environments to conduct experiments, both in philosophy and social sciences. Section 1.3.4 contains an overview of some of these researches. The difference and similarities with my approach are outlined. In this way the readers are, hopefully, in a better position to understand the approach taken in this thesis.

1.3.1 Intelligent and moral agents

(Moor, 2006) refers to machines as agents with physical and computational aspects. He argues that computational activities have a moral dimension, as there are always various ways (including wrong ones) in which things / situations / activities are computed, analysed, etc. Key to his argument is that machine ethics needs to "move beyond simple normativity to richer senses of evaluation." For varying situations we will need varying richness of morality. Different applications will require different complexity in moral awareness and reasoning. Moor sketches a continuum of increasingly rich moral agents:

- Ethical impact agent

- Implicit ethical agent
- Explicit ethical agent
- Full ethical agent

The first type 'happens' to be ethical by the impact it has on our lives through its very existence. An electronic signal transmission tower was not designed with any explicit ethical considerations in mind. Its goal is very practical and instrumental to relay and amplify transmission signals. Because there might, for example, be potential health impacts it gets a moral dimension that was never intended, not implicitly nor explicitly. For the next type holds that the way the machine functions is designed such that unethical actions or functioning is not possible. This is the implicit ethical agent. The explicit ethical agent has the reasoning about the ethical aspects of its functioning designed into it. The explicit ethical agent is designed such that not all its deployment situations and decisions are foreseen, but left to the agent to reason about them on its own initiative. The final degree of ethical reasoning brings the agent to the human level of moral awareness and reasoning: the full ethical agent. Though Moor does not exclude a priori the possibility of the latter category it appears to be a long way out. His focus is more on the explicit ethical agent, which is challenging to develop but within the realm of possible within the foreseeable future. It is this kind of ethical agent that I focus on. Two chapters, chapters three and four, of this thesis are explicitly addressed at trying to understand what it takes to develop some sort of autonomous moral reasoning capacity in a software agent. This lacks obviously the physical embodiment of a machine. This a clear limitation of the scope of this research. Developing software that can be said to contain some sort of moral reasoning capacity in its own right is, however, already a daunting challenge. The lack of embodiment does away, at least for this research, with questions of perception, spatiality, etc.

1.3.2 Moral responsibility and artificial agents

When we build artificial agents and let them act the question arises who is responsible for their actions and the consequences. Can these artificial agents be held accountable? If not, who is accountable? The first question is whether moral responsibility can be assigned to non-humans? If not, the issue is settled. Most people and many researchers have left the question open, or answered affirmative. If the possibility of morally responsible artificial agents is not ruled out a priori, the next question is under what conditions, to what extend, and with which understanding of responsibility? Stahl (2006) suggests, that we can to some extend, if the notion of responsibility is adjusted, or a new notion is introduced (quasi-responsibility). Adam (2005), drawing on the work of Floridi

(2004b) concludes also, though on different grounds, that we can delegate some moral responsibility to artificial agents.

In the context of this research the question is what the impact of this debate is on the research in this thesis. This research is looking into the possibility of creating artificial agents with moral reasoning capability. This does not imply the actual use of these agents, nor does it automatically entail that, if such capability can be constructed, responsibility should also automatically be assigned to these agents. In this sense this research precedes this discussion. If artificial agents cannot, in any meaningful form, reason about morals it seems to make little sense to hold them accountable. More importantly, the primary aim is to construct artificial agents in order to experiment and help theorizing. This is an activity in which the agents have no impact for which they can be meaningfully held morally accountable.

If they are applied, as is tentatively suggested in the chapter on deontic constraints on informational relationships, the question still is whether they can be meaningfully held responsible. Here I maintain, that as long as the agents constructed are explicit ethical agents, in the sense described in the preceding section, the discussion of responsibility is the same as on moral responsibility for general computer systems, robotics, etc.

On both of the above accounts the topic has been descoped from this research. Though for further future research on the delegation of moral responsibility, it might certainly have some to contribute. If we can build artificial agents that can perform the right actions and give a proper moral justification for those action, we might be much more inclined to actually delegate that responsibility. But first, the outcomes of this research need to be assessed.

1.3.3 Rule-based ethics, connectionism, universalism, particularism

In this section I will refer to some meta-ethical notions and the connectionist debate to clarify my approach. First, there are two choices that guide most of the approach. One, I will use the belief-desire-intention model proposed by Bratman. All the implications of this choice I accept, and will not try to defend. The BDI-model has been widely used and tested. And though there are clear advantages and disadvantages, I feel justified to refer to the BDI-model as a widely accepted model. Two, I try to be as neutral towards various ethical and meta-ethical positions as I can within the limits implied by the choice of the BDI-model. I hope that my approach will suit or be usable to utilitarian and to the Kantian, to the non-naturalist and to the naturalist moral philosopher.

Do moral propositions represent universal moral truths or are they relative to a particular culture or historical period, etc.? With logic and universal quantification universalism is just around the corner. And, as my approach is based on

logic as its modelling component, it might seen to have universalist claims. The aim, however, is to provide the tools to express moral convictions irrespective of their claims of applicability. Scope and range of application needs to be determined separately in practical interpretation of the formalism. In this sense I think my approach is indeed neutral. My underlying assumption (which I try to validate) is that all morality can be expressed using modal logic. In this different sense my approach can be called universalist. I cannot exclude the possibility, however, that some moral propositions cannot be expressed with the tools I propose.

Another underlying assumption of this thesis is that we talk about moral values, and that it makes sense to talk about them in the context of artificial entities. The source and justification of particular moral values is, however, irrelevant.

Also how an artificial agent perceives is outside the scope of this thesis. The interest is in the *structure* of the moral reasoning. Thereby the source of the value, its nature and justification are abstracted from.

From the cognitive science and philosophy of mind research areas comes the distinction between the connectionist approach and symbolic, classic, approach. They are often depicted as opposing approaches towards the modelling and understanding of mental activities.

According to connectionism mental phenomena can be modelled as network of connected units. The units are simple in the sense of low-dimensional input-output constructs. The well known example is that of a neural network in the brain where neurons are connected through synapses. The network connections are dynamic. The strength of the links is increasing/decreasing in time depending on the activation of the links and the propagation mechanism.

The symbolic approach depicts mental processes as symbolic manipulation or as computational activity, basically a Turing machine. The syntactical rules governing the computations are main focus, together with the structure of the symbols representing the mental state. The symbolic approach is rule-based[11]. A part of the debate between the two approaches focuses on what is a proper representation of, or which model is closer to the neurological structures in the brain, and which of the processes (rules vs forming through environmental stimuli) provides a better model of mental activity.

Since the basic modelling component in this thesis is modal logic it can be easily seen as siding with the symbolic approach. That would be too strong a position. In my view

a) it is not necessary to construct anything resembling a brain;

[11] It is important to note that rules do not imply predictability, nor that non-rule based (e.g. stochastic) systems imply unpredictability.

b) symbolic representation of mental activities can provide a useful model even of the brain;
c) symbolic representations can be combined with networks to create models of higher abstraction level.

Ad a) In this thesis the assumption is that, for artificial morality it is not necessary to construct anything that can lay claims to direct resemblance to the human brain. Machines and software agents are the main focus of this research, and have totally different physical make-up.
Ad b) For different purposes of analysis different levels of abstraction are required in research. The behaviour of gas can be described by the Ideal Gas Law (relating volume, pressure and temperature) and by the Kinetic Theory of Gases (describing the behaviour of individual molecules) (Jansen, 1990) . The same goes for macro-economic and micro-economic theories. In this vain symbolic representations of mental activities at times can be very useful (imagine the extensive set of neural nets one would have to model to represent moral propositions!). At other times it can pay to investigate the lower level processes as modelled in networks.
Ad c) In this approach a middle position is taken between the symbolic and the connectionist approach, in that the activity of agents is constructed as a set of plans (a set of actions) that are connected through events: plans send and handle events. These plans contain higher abstractions that constitute the units in a neural network, but are still fairly simple.

1.3.4 Experimentation in philosophy and social sciences

Experimentation with theories using computer models is being done in both philosophical and social sciences domains. As illustration of the possibilities this section provides an overview of some experimental approaches and architectures in different domains.

Negotiating agents

(Dumas, et. al., 2002) have developed an approach for the modelling and execution of negotiating strategies. Their motivation is the increased amount of commercial transactions through the Internet. As a consequence the interest in and the use of software agents as a technology for automated negotiations has increased. Their aim is to provide a framework in which various strategies and auction mechanisms can be tested through the actual execution. For the modelling aspect defeasible logic is used (Dumas, et. al., 2002, 2).

> "...our aim is to develop a simple yet expressive framework for specifying negotiating agents' strategies, in a way that their decisions are predictable and explainable. Specifically, we explore the suitability of defeasible logic

> programming for expressing the decision-making process of negotiating agents, coupled with statecharts for expressing their internal coordination."

The approach consists of the following steps. A negotiating strategy is modelled using defeasible logic. The choice for defeasible logic is driven by the possibilities for non-monotonic reasoning. The modelled strategy is translated into UML statecharts. Statecharts are finite state machines describing various states and the conditions for the transition from one state to another. The state charts are then coded manually in executable software. The authors point to the possibility of having the statecharts compiled automatically into executable code. The architecture of the executable software framework is the following.

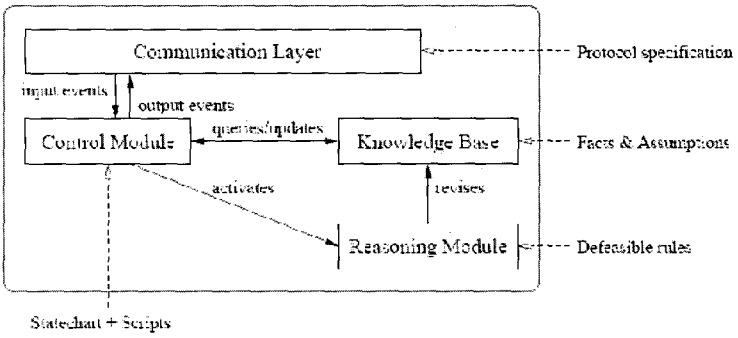


Figure 1: Negotiating agents architecture

This approach is elegant and powerful. There are some similarities and interesting differences with the approach followed in this thesis. The subject matter is obviously a different one: morality versus commercial negotiating transactions. There are, however, more similarities than differences. Both approaches use logic for modelling purposes. The choice of logic is different however. As the topic of this thesis is morality, deontic logic is the most obvious choice, since it is specially designed for the purpose of reasoning about obligations. The interest for Dumas is in strategies in which uncertainty, and incomplete knowledge about what to expect from the other negotiating agents is the key aspect. From this point of view defeasible logic is a natural choice. This does not imply that defeasible logic is not suitable for the approach in this thesis. The goals are different, however, and that accounts for the different choices. Limitations of scope have led to the exclusion of addressing the issues that might require looking into defeasible logic. As yet, the DEAL framework is expressive enough to capture all notions relevant to this research.

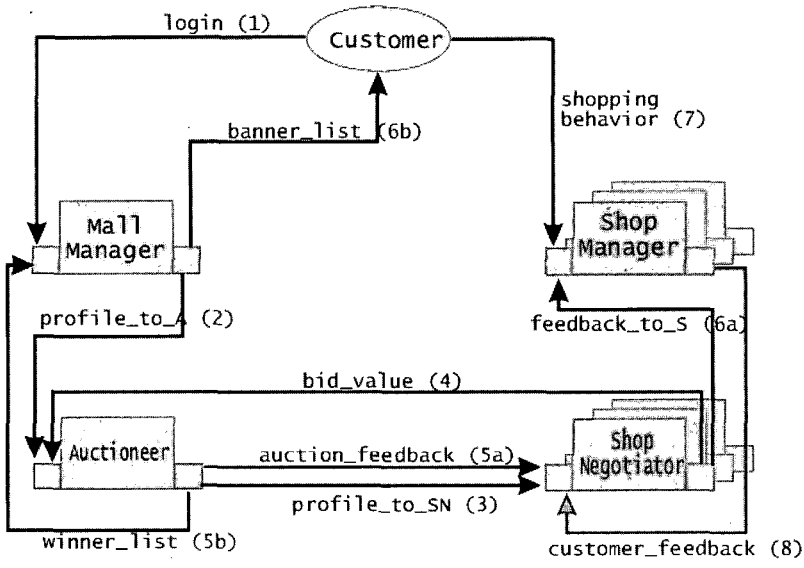


Figure 2: Extensible agent architecture

An Extensible Agent Architecture

('t Hoen, et. al., 2002) have developed an agent architecture not unlike Dumas (Dumas, 2002). Again the focus is on economic, transactional aspects

> "With the advent of electronic marketplaces, scale limitations as encountered in the brick-and-mortar world no longer apply. At the same time, novel problems are encountered, like how customers can find their way in large marketplaces. A currently preferred solution is to have a central party propose relevant suppliers and products to a customer. This central filtering mechanism uses knowledge of users, shops, and the product domain to determine recommendations. Such recommender systems work well within limited domains, e.g. a book or music store. However, to maintain accuracy in a large marketplace with many customers and suppliers, the latter will need to reveal detailed and perhaps sensitive business information to the central party. Central filtering-mechanisms may thus suffer from objections from the suppliers. Furthermore, the amount of information to be processed and maintained by the central party can become unmanageable. Other approaches are needed to complement centralized systems." (' t Hoen, 2002, 2)

The implementation is geared towards economic transactions of a particular sort. It is not suited for general purpose use, of varied domains of application. As such, the framework used in this thesis is more generic. The modelling of the

behaviour has a strong mathematical orientation, combined with first-order logic.

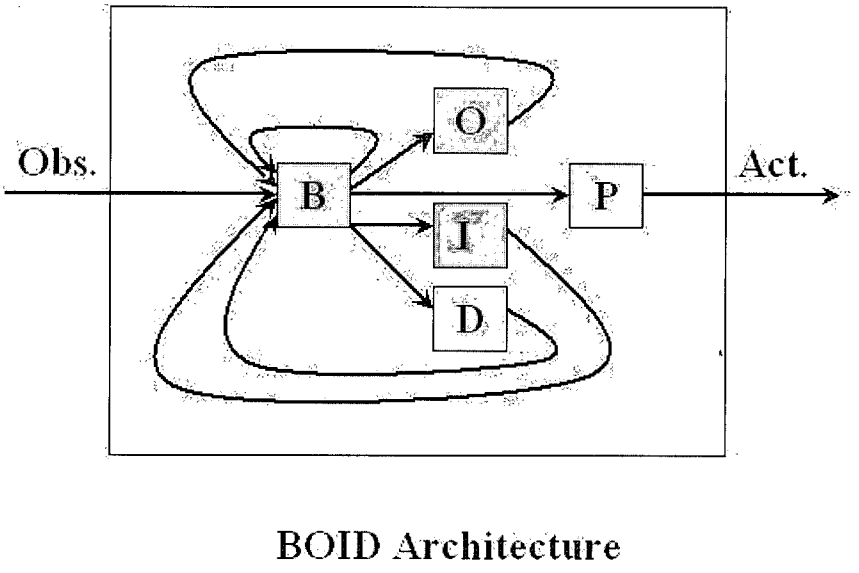


Figure 3: BOID Architecture

BOID

BOID is "an architecture for Beliefs, Obligations, Intentions and Desires." (Broersen et. al., 2001a). The central component is a mechanism to assess the impact of an action before committing to it, and to resolve conflicts between the four components. The four components represent the

> "...mental attitudes of an agent
> belief: information about facts, and about effects of actions
> desire: internally motivated potential goals
> obligation: externally motivated potential goals
> intention: committed goals "

The behaviour of these components, understood as input-output processes, is governed by production rules.

The BOID architecture is in the first place a conceptual architecture for reasoning about various modalities. The implementation in software is of secondary importance, though some executable Prolog implementations are available.
This approach and my approach share the use of modal logic as a central component. However, in the BOID approach the focus is on the formal aspect of modelling in logic. Whereas in the approach of this thesis modal logic is used primarily as a modelling language for implementation purposes. Related to this difference is the difference in implementation.
Dastani and Van der Torre (2002) argue in favour of the BOID architecture over the BDI framework by reference to the supposed limitations in the implementation phase of the software development.

> "Although these formal tools [Rao and Georgeff's BDI framework, VW] and concepts are very useful to specify various types of cognitive agents, they are specifically developed for the analysis phase which makes them too abstract for other phases.O Dastani and Van der Torre (2002:2)

They ignore the work that has been done since the specification of the BDI framework since the publication of the BDI framework (Rao and Georgeff, 191, 1992) . Wooldridge (2002), Agent Oriented Software Pty, Georgeff (1998) and Guerra-Hern·ndez (2004) among others, have extended and implemented the BDI framework to such an extend that it can lay claim to be one of the leading frameworks of combined analysis, design and implementation. The BOID implementation is limited in its agent aspect. Existence across distributed networks, inter-agent communication, teamwork, for example, are not supported, or not as sophisticated as in the framework used in this thesis. The approach defined by Bresciani et. al. (2002), Tropos, shows an approach where it is possible to create a framework for all phases of the development. Tropos is based on the BDI framework, and uses the JACK agents development environment. Though it certainly has short-coming still, it is among the more mature approaches available. Dastani et. al (2004) note in later work that "Of all methodologies considered, Tropos comes closest to a complete development methodology for multi-agent systems." Tropos and the approach in this thesis are for all practical purposes the same. The only difference is that for modelling purposes the module that ships with JACK is used rather than an external modelling tool. This has the clear advantage that the design can be compiled into code semi-automatically.

Animated Specifications of Computational Societies

Artikis, Pitt and Sergot are interested in the constraint on action in artificial societies. In open agent societies non-confirmation to the rules of engagements is a real issue. The modelling of the rules, their enactments and punishment of violations will be a key in the functioning of many applications in the economic and social spheres where humans and computational societies interact.

> "E-markets and digital media rights management are examples of application domains where software agents form computational societies in order to achieve their goals. Key characteristics of such societies are agent heterogeneity, conflicting individual goals, limited trust and a high probability of non-conformance to specifications. Consequently, it is of eminent importance that the activity of such societies is governed by a framework with formal, verifiable and meaningful semantics. We address such a requirement by presenting a formal framework for specifying, animating, and ultimately reasoning about and verifying the properties of open computational societies/systems, i.e. systems where 'the behaviour of the members and their interactions cannot be predicted in advance'." (Artikis, 2002:1)

The social constraints on actions are approached from three angles: validity (agents hold the power to execute the action); permissibility (the deontic angle of detailing obligatory, permissible and forbidden actions); enforcement (sanctions on invalid and/or impermissible actions). The state of a society is described by the institutional powers, normative constraints, roles, and communication language. The normative aspect is modelled through the deontic operator O, 'it is obligatory that'. Reasoning about events and actions is done using the Event Calculus (EC).

> "It is based on a many-sorted first-order predicate calculus. Events (actions) in EC initiate and terminate fluents, which are properties that have different values at different points in time. The value of fluents is affected by the occurrence of events. A fluent starts to hold after the occurrence of an event that can initiate it. Similarly, a fluent ceases to hold after the occurrence of an event that can terminate it." (Artikis, 2002:2)

For the execution Artikis et. al. use a computational framework called the Society Visualiser, which contains components to process the events, to compile the new social state after the event (using the constraints, institutional powers, etc.), which is displayed via the visual representation component. The society is depicted below from (Artikis, 2002, 7). The execution basis is made up of the

Prolog programming language with a database to store social states and the society specification.

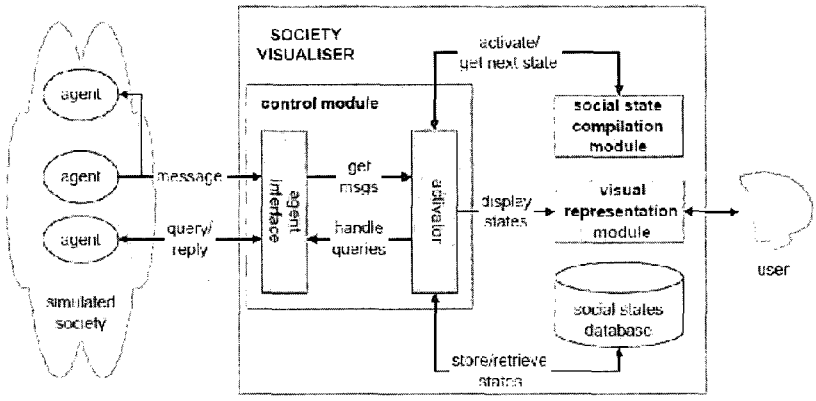


Figure 4: Society visualiser

The Event Calculus as modelling component is very powerful in the approach of Artikis et. al. It is as expressive as the DEAL framework. In some respects it is more specific, e.g. in specifying the roles. In other respects it is somewhat less pronounced, notably in the epistemics of the agents, e.g. the handling of beliefs. The implementation is less well developed than in JACK. The internal workings of the agents are not specified. This is probable due to the different aim: validating the constraints on agent systems rather than building the agent systems. As such it is complementary to the approach presented in this thesis.

1.4 Organization of this thesis

1.4.1 Background

The development of the personal computer has introduced both new fields of philosophical research, and of tools for philosophical research. It appeared to me, that by using these tools, a laboratory for philosophical research could be constructed. All kinds of constructs can be made to test and investigate philosophical theories in ways that were never possible before. I constructed a computer model of utilitarian agents that were engaged in voting. The experiment was

meant to check the validity of various claims made in the debate between rule and act utilitarian theorists. This resulted in the first paper, here chapter 4.

Of course, I soon found out that I was not the only one to come up with such an idea. Reading more extensively I found that several, though not yet many researchers, were actually building artificial constructs. I read extensively, and compared the work of various researchers. I wondered about the approaches and the apparent lack of a formal method. I tried to formulate a methodology that underlies some, and hopefully much, of the research being done. This provides at least a descriptive methodological framework, and possibly a prescriptive one. This is the first chapter of my thesis.

From the methodological considerations followed a clear need for a modelling framework. This framework I found in the DEAL (deontic epistemic action logic) framework. Later to be extended with the belief-desire-intention model. This framework allows me to model most relevant moral notions needed to experiment with moral theories. I also wondered at the great effort involved in coding all required software. As the agent paradigm seemed very well suited to my purposes I looked around and found an industry strength software package, JACK, that provided the framework for the programming of software agents without having to do all the basic work. Which would have been impossible, but even more, not the key purpose of a philosopher. Though programming / software design as such seems, to me, to be an important skill for a philosopher who wants to push his investigation further.

Using these two tools, the modal logic and the agent software, I started investigating the possibilities to provide software agents with a moral reasoning capacity. Taking a general, neutral view of the moral debate I investigated what is, and what is not possible. I found that, as a general toolkit, these two components will get me a long way towards my goal. It becomes, however, also very clear, that it is not possible yet to implement a general purpose moral reasoning facility for software agents. Moral reasoning is too complex, and the epistemology involved too complicated, to achieve any such thing in the near future.

Special purpose moral reasoning, in contrast, seems feasible as is shown in chapter 3. Informational relationships are deontically constrained. It proves possible to construct software agents that can reason at relative sophisticated levels about obligations to protect privacy and obligations to inform persons.

I coined the term *SophoLab* as a catch phrase for both the idea of experimenting in philosophy, and the methodological framework with its techniques and tools for modelling and implementation. It is a close kin to its older sibling 'The Computational Philosophy Laboratory' (Magnani, 1994-2006) , and its cousin 'Home of the BOID' (Broersen et. al).

1.4.2 Hypotheses and research questions

The body of this research is made up of four chapters. Though not originally conceived as one whole but written as separate papers they sprang from the same underlying ideas and assumptions. Hence I feel that, in retrospect, I can still formulate the hypotheses and research questions without being artificial.

The hypotheses underlying the current research are the following:

1) there is a common, underlying methodology for experiments conducted in the field of computational philosophy;
2) experimentation more specifically with *ethical* theories is possible and fruitful, i.e. increase understanding of a theory, enhances the insights into the implications of a theory and provides support for a theory;
3) DEAL / modal logic and software agents are well suited for the purpose of experimentation and application, and allow the capturing and implementation of thin, central moral notions such as obligation, right and permission.

The hypotheses can be detailed by the following sub-questions:

1) is there a common underlying approach for the various philosophical research conducted with the help of computers?
2) what does a methodological framework for the use of computers in philosophical research look like?
3) what are the methodological guidelines for researchers involved in experiments in the field of computational philosophy?
4) is the combined modelling capability of the BDI-model and the DEAL framework sufficiently rich to capture central moral notions?
5) can the logical models be implemented using the JACK agent language to provide an environment in which the model can be executed?
6) is it possible to capture, express and implement deontic constraints on informational relationships?
7) can experimentation shed light on long-standing debates in ethics: i.c. the utilitarian research community about the comparative strengths and weaknesses of act and rule utilitarianism?

From these hypotheses the following research goals and deliverables are derived:

1) the definition of the methodological framework underlying the experiments conducted in computational philosophy;
2) the definition of a modelling framework for experiments in moral philosophy;
3) the development of a software tool set that allows the implementation of the moral models in executable code;

4) to illustrate the plausibility and viability of deliverables 1 through 3 via the conduction of experiments.

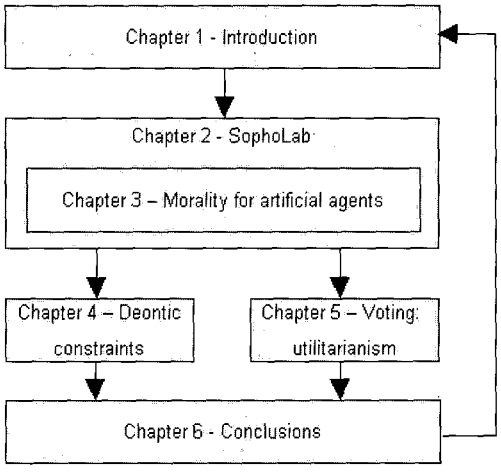
1.4.3 Thesis lay-out

This thesis is organized as follows. Chapter 2 provides an overview of the field of experimental, computational philosophy. It focuses on what it means to use computers to create test environments for normative theories. It is based on various research done in the field of computational philosophy, and in this thesis. It contains a case study from moral philosophy (Danielson), and from philosophy of science (Thagard) to illustrate the likeness in approach. This part is written from the perspective of an observer, noting what is going on. It reflects on the current practice. As such, it is a descriptive methodology, and not a prescriptive methodology. In this reflection several methodological issues are addressed.

Chapter 3 defines several basic moral notions and concepts, and models these using DEAL. It outlines the programming toolkit, JACK software agents. An experiment is set up to gain a better understanding of the nature of negative moral commands ("thou shall not..."). In the process of constructing artificial agents with an ability to use negative moral commands it becomes clear that our understanding of negative moral commands might require revision. Also in this chapter, the strengths and weaknesses of the approach are investigated. By asking what it would take to provide artificial agents with moral reasoning capacity it pushes (beyond) the limits of what is possible with current technologies, and our understanding of moral reasoning.

Chapter 4 contains a discussion that fits better under the header of applied ethics. It uses the tools developed for experimentation to suggest practical solutions to moral challenges. It shows that the tools and techniques used to investigate, and empirically support normative positions, can also be used to apply these normative positions to the very same practical situations. Chapter 4 emphasizes more the use of DEAL and agent software. In a limited setting, it proves possible to both model and implement moral notions in such a way that practical application is not a mere theoretical possibility. Using DEAL the deontic constraints in informational relationship are modelled and implemented. Agents representing users in an abstracted but relevant case share information, and can reason about the protection of privacy.

A major driver for the development of the approach and experiments in this thesis is complexity. The subject matter of normative ethics is a world of a great many relationships. In order to derive empirical support for normative positions, practical cases with their inherent complexity must be investigated. This is hard to do without computerized support. From the perspective of application it is the



Chapter 1 sets the scene, defines the themes, research areas and hypothesis
Chapter 2 defines methodology, provides overview of computational philosophy
Chapter 3 develops tools and techniques for modelling and implementation; defines limitations to approach
Chapter 4 demonstrates viability of approach in limited applications domain. Focuses on deontic constraints on information relationships.
Chapter 5 idem on utilitarian debate which is illuminated through experiments.
Chapter 6 revisits the hypotheses and research

Figure 5: Thesis lay-out

same complexity that drives the need for computerized support. The consistent application of particular norms throughout a complex situation is not possible without computerized support either.
A long standing debate on rule and act utilitarianism is the subject of chapter 5. There is a tendency to prefer rule utilitarianism over act utilitarianism on the ground that there is a host of situations in which act utilitarianism is not able to provide a strategy to groups of agents that solve situations in which coordination is required to achieve a particular goal. The arguments rely on apparently sensible examples and representations of the behaviour of agents. Chapter 5 shows that 1) through implementation of a theory, its missing elements can be identified more easily, and 2) running a theory in many different configurations can provide unexpected conclusions, that through the complexity of all the elements working together could not (as easily) have been derived via armchair philosophy.
In chapter 6 the goals and research questions, described in this introductory chapter, are revisited. I draw some conclusions and evaluate the results, the answers found and not found.

2 Methodological reflections on philosophy and experimentation

Abstract

Leaving the armchair, philosophers are increasingly turning to computers and other means to test their theories and gather data to support their theories. The introduction of experiments has opened a whole new field of research. There are many initiatives that differ in tools and techniques used, and in theories tested. Nonetheless it is my claim that many of these works share a common underlying approach. In this chapter I sketch the outlines of a methodological framework that underlies the research of many philosophers working in this new field. First, I illustrate the methodological framework by analysing two paradigmatic cases of forerunners of this research field: Danielson and Thagard. Next the methodology is discussed in more detail. Then some methodological issues associated with experimentation are addressed: to what extend does the choice of technology and conceptual framework influence the outcome of the experiments; what standards should the researcher adhere to, etc. Concluding I claim that my proposed framework is descriptive of the field. I leave open the question whether it should be prescriptive.

2.1 Introduction

"From time to time, new movements occur in philosophy" and computing is the onset of such a movement, according to (Bynum and Moor, 1998, 4). The work of Sloman (1978) , Danielson (1992, 1998), Pollock (1995), and the philosophers collected in the books substantiate this notion. The terms "computational[1] philosophy" (Thagard, 1988) and "philosophy of information" (Floridi, 1998) are often used to denote philosophic research activity involving computers. Computing figures here in at least three different roles: as subject matter, method and model. Computers as subject matter refers to the role computers have come to play in our lives and in society. As method it refers to computers and computer models as tools for research. Computers can also function as a

[1] Not all referenced authors use the term computational philosophy but they stress the computational aspect of the new research.

mould to shape and test ideas, and as such function as a model. For computing as method the experimental nature that is the key aspect.

Many of the philosophers collected in the above mentioned publications focus on the use of computers to model research questions, and to investigate them through simulation. Each of them is developing his own equipment to express theories in such a way that they can be implemented in an artificial (non-human) environment. In doing so a laboratory for philosophical experiments is being created. Let's refer to it as SophoLab. I claim that several of the research currently being carried out has the same underlying methodology[2]. I will describe this methodology and illustrate it by reference to the work of Danielson and Thagard, who provide some paradigmatic examples of experimental (computational) philosophy. I have chosen Danielson and Thagard because they have as one of the first in their respective fields, and very extensively, introduced computer based experiments. More than most researchers in the earlier days they have gone quite a long way in actually building computer models of their theories, and executing them as experiments. Both also feature in the first overview publication by Bynum and Moor. I do not claim that they are the perhaps the best examples, let alone the only ones. In the field of philosophy of science Lindley Darden (1991, 1998) would have been a good alternative. Walter Maner (1995, 2002) has written on ethics and engineering focussing on computer programs to reasons about ethical dilemma's. But neither he, nor the authors cited in his 2002 article, go to the same length as Danielson in constructing computer models and using these to actively support theorizing.

2.1.1 The role of the experiment

The use of new methods from outside the field of philosophy is advocated strongly by Danielson. In his research of game theory in relation to morality he advocates actual construction and testing of ideas.

> "I propose that we actually build the agents proposed by the contending theories and test them instrumentally. This promises to solve questions of coherence and efficiency of moral agents." (Danielson, 1992, 17)

[2] In this chapter only two case studies are referenced. In addition to the experiments in the others chapters, this constitutes the only formal reference made. Much more research has been studied. Limits in time and scope have prevented me from presenting more cases. Though the methodological considerations presented in this chapter are broadly based no claim to universality is made. Also, the methodology is descriptive and not prescriptive in nature.

Danielson not only sees this as just a means of improved theorizing, he actually regards it as a necessary exercise: "...I find that verbal arguments do not suffice; it is unusual to claim that computers are necessary to providing a justification of morality.", (Danielson, 1992, 17). Artificial morality, combining game theory and artificial intelligence, as proposed by Danielson, is a paradigmatic example of experimental computational philosophy. Bedau (1998) very much supports this point and indicates that the inherent complexity of the subject matter is too great to solve without experiments. His experiments are based on insights from artificial life and implemented using computer programs[3].

> "These are 'idea' models for exploring the consequences of certain simple premises. Artificial life simulations are in effect thought experiments ñ but *emergent* thought experiments. As with 'armchair' thought experiments familiar in philosophy, artificial life simulations attempt to answer 'What if X?' questions. What is distinctive about emergent thought experiments is that what they reveal can be discerned only by simulation; armchair analysis in these contexts is simply inconclusive. Synthesizing emergent thought experiments with a computer is a new technique that philosophers can adapt from artificial life." (Bedau, 1998:145)

What we witness increasingly is the use of the theories, originally developed to better understand particular phenomena, in constructing new means of philosophic research. Thus emerges what I would call "recursive philosophy": philosophers involved in the research of artificial intelligence bring their traditional equipment ñ philosophical methodology and theories ñ and apply those to the new field of research. We now see how the results from this new field are being brought back and are being used in philosophy itself, and provide exciting tools for philosophical experimenting.

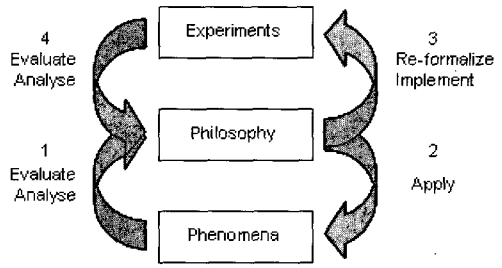


Figure 6: Experimental, and sometimes recursive philosophy

[3] Though not mentioned by any of the cited authors the formalization of theories and the implementation on computers also offers the ability to compare and evaluate competing theories by criteria such as decidability and provability. It was Jan van den Berg who first pointed out this opportunity. It is addressed in section 2.9.

Phenomena provide input to the process of analysis from which springs (the beginnings) of a theory (1). A theory that is used to reflect on the reality (2), to put a structure to the way we look at, and interpret 'reality'. A theory can also be applied to the reality. For example, to achieve something, in the way that the laws of physics are used to design airplanes. The results of this application, and of the predictions made, provide also a reference point for evaluation purposes and possible refinement and reformulation of the theory (1). The theory can also be tested by constructing experiments in a lab environment (3). These experiments in turn provide new evaluative material to assess the correctness of the theory and propose new theories (4). Though this is a simplified picture of the very complicated processes in which phenomena and theories, and theories and experiments, interact it will serve as a base reference for the discussions in this chapter.

Van den Hoven and Lokhorst (2002:8) illustrate the mechanism described above.

> "In addition to the more traditional methods of moral inquiry into the issues, several attempts have been made to utilize computer programs and information systems to support moral reasoning and help us understand moral behavior. If these attempts were to be successful we would be presented with an extraordinary full circle: computer technology would come to the aid of those grappling with the moral problem, to which computer technology itself has given rise: computer supported computer ethics."

'Computer supported computer ethics' is the key phrase. Computer ethics is the original field of research. The research is conducted by means of 'the more traditional methods of moral inquiry', but eventually the very same computer comes to the aid of the researcher. The fact that the means of research are subject of philosophical research itself accounts for the recursive nature of the philosophising in question.

Paul Thagard (1988) has used computer models to research the structure and growth of knowledge.

> "At the core of epistemology is the need to understand the structure and growth of scientific knowledge, a project for which computational models can be very useful." (Thagard, 1998:48)

This movement is, from a philosophical point of view, a radical change. It implies that philosophical theories have their theorizing informed by patterns generated by computer programs. This would make available a whole new range

of tools like neural network analysis, data mining, statistical analysis and pattern recognition into the field of philosophy.

All the above cited authors emphasize one, and often more, of the following characteristics as being important in the approach.

- Inherent complexity requires new means of research
- Inductive reasoning through and with the help of simulations
- A high degree of interaction between different entities in the theory
- Intensive use of formalisms (logic and/or programming languages)
- The rigour required by modelling is seen as a clear help

Behind these characteristics and objectives in research there is a similar way of proceeding in research. I will attempt the definition of a programme in which several aspects of the use of new techniques to aid philosophy will be coherently defined and accounted for. The outline of the program is not a clear, well-defined objective. It is an attempt to sketch the outlines of experimental philosophy.

A programme of experimental philosophy will be successful if it helps in clarifying or refuting existing theories, and suggests new avenues of research[4]. There are many ways in which this can be brought about. An experiment may draw attention to unforeseen consequences of particular assumptions, and point to incompleteness or inconsistencies that come to light in the attempt to model the theory in order to fit the experimental settings.

[4] This thesis is not a work of philosophy of science though it contains ample reference to criteria on which a theory can be judged. These criteria are exogenous to the work of experimental computational philosophy. Reference is made to criteria, and even discussed at some length, without the intention to take a position. Mostly references will be made to criteria that are commonly accepted as indicators of, or minimal conditions for, a good theory, e.g. consistency, power of prediction. Ultimately, it depends on the researcher and the research community which criteria are used. Experimental computational philosophy will be instrumental in evaluating the theory based on these criteria, in particular through offering new means to express and measure the criteria. (see section 2.9)

2.1.2 Reality, theory and computational model

In the previous section, I described how based on a theory, abstracting from details, experiments can be set-up to test and further investigate the theory, by comparing the predictions of the theory and the outcome of the experiment. The theory in turn can be applied to reality in, for example, a prescriptive or predictive manner. In experimental philosophy, and in this thesis in particular, the emphasis is on the relationship between theory and experiment. How do philosophers proceed when setting-up experiments? What are the requirements and prerequisites for experimentation? These are the questions that drives this research. How the observed phenomena and theory fit together is not the object of this research.

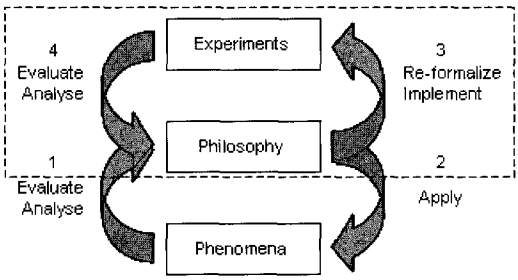


Figure 7: Focus of experimental philosophy

2.1.3 The organization of this chapter

In the section 2.2, 'Methodology of philosophical experimentation', I will define the program and the methodology. In conducting experiments the translation from theory to experiment is of crucial importance. This translation is facilitated through the use of intermediate conceptual frameworks. Because the methodology, and the use of conceptual frameworks, that is described, is not trivial or directly apparent I will discussed first, two paradigm cases of research in this area. These are the use of a computer model ECHO by Thagard (section 2.3) to investigate the transition from one scientific theory to another, and, the studies in artificial morality by Danielson (section 2.4). These, together with section 2.5 which provides a short reflection on Danielson and Thagard, will help the reader understand the exposition of the methodology that follows in the next sections. First, in section 2.6, a detailed description is provided. Section 2.7 zooms in on the key element of the methodology: the conceptual framework. In the following section (section 2.8) particular methodological issues in conducting experiments are discussed. E.g. what are the standards for the verification of results? The

formalization inherent in many of the approaches has an interesting side-effect. It allows for the use of additional criteria to evaluate the theories in question. This is addressed in section 2.9. Finally, in the last section (section 2.10), some conclusions are drawn on the usability of the framework.

2.2 Methodology of philosophical experimentation

2.2.1 Research program and vision

The use of experimentation and simulation is fairly new in philosophy. I argue that most authors have chosen a similar approach. This approach, the implied values and the objectives allow us to draw the contours of a programme of research centred around experimentation. In this section I first try to define this program. Next I describe the methodology that forms the core of the experimental research. All of this takes place in the experimental 'world', as depicted in figure 7.

A research program
Experimental philosophy can be seen as consisting of four elements:

- a program and vision ñ directing all efforts
- a methodology ñ linking theories and experiments
- a laboratory ñ an environment containing tools and techniques to conduct experiments
- a set of techniques ñ instruments in conducting experiments

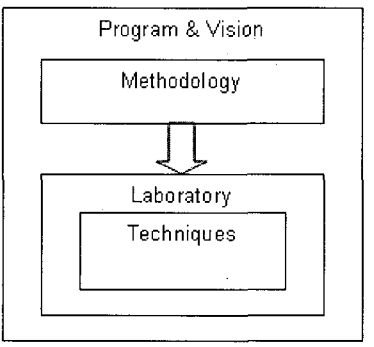


Figure 8: Experimental philosophy

The laboratory is the container for the techniques. The deployment of the techniques and the configuration of the laboratory are determined by the methodology. The program and vision determine the direction of the whole as well as its

boundaries: what is a legitimate subject of experimental philosophy and what is not.

Vision statement

Technologies developed in other sciences and arts can be used to express philosophical theories in different vocabularies[5]. These other vocabularies allow for the construction of experiments in which a theory can be investigated, I hope to demonstrate, more intensively and from other angles than the traditional philosophical methods allow for.

It is possible to construct a laboratory of philosophy, the SophoLab. SophoLab will consist of a collection of techniques for conducting experiments with philosophical theories. Underlying these techniques will be a methodology defining how to proceed in experimentation.

I propose to define experimental philosophy is an approach within philosophy dedicated to the development of a methodology and of techniques that will allow the conducting of philosophic experiments. Philosophical experiments will further the investigation of theories from new angles and in ways that are not accessible by mere theorising and conceptual analysis. E.g. moral theory concerns itself with human interaction. Human interaction, especially within and between large groups, can be so complex that it is impossible to foresee all outcomes and investigate them by 'mere' reasoning. Only through conducting experiments do we get a view on the wide range of possible outcomes.

Domain of experimental philosophy

- Experimental philosophy will concern itself with the means and methodology of 'translating' theories to experiments, step 3, devising experiments (abstracting theories into experiments) and implementing in figure 7; conducting experiments and the standards that apply to experimentation, constituting the experimental 'world';
- and the translation back from the experiments to the theories, as part of which
  - Experimental philosophy will state the extent to which a theory is complete and can be subjected to experimentation. This is in itself not a statement about the adequacy or correctness of the theory. Though incompleteness certainly does raise some questions. It also suggests new

[5] By vocabulary I mean the concepts and terms of the technologies developed in other sciences. E.g. an moral theory can be reformulated in the terms and concepts (the vocabulary) of game theory, which in turn allows it to be reformulated in the vocabulary of the Java programming language. A more detailed discussion will follow in this chapter.

theories or adaptations to the existing theory. This is the evaluative and analytic step 4 in figure 7.

- Experimental philosophy will analyse the results of the experiments with regard to the compatibility with the theory under examination. This is the evaluative step 4 in figure 7.

- Experimental philosophy in a strict sense will not be concerned with questions like how to adjust the theory in order to accommodate the outcome of the experiment. This is part of theorizing. In practice, they might go hand in hand, and be performed by the same researcher. But they are two different activities nonetheless.

Experimental, *computational* philosophy is defined as a sub-field of experimental philosophy that uses as its tools and techniques for experimentation exclusively computer and computer programs.

2.2.2 Methodology

The essence of experimentation is to transform a theory into artificial constructs and working conditions in a controlled environment. The controlled environment resembles reality while, at the same time, abstracts from it, to eliminate less relevant aspects. Theories are investigated by manipulating the artificial constructs. The theory is transformed into an entity, an artefact, that is executable, something that can be run on a computer. The operative word is *running*. The methodology of experimental philosophy is about getting from the one, the theory, to the other, the functioning computational artefact, so that it can be executed, or done. Transforming, or translating, is deconstructing the theory into smaller elements that can be reconstructed in the new environment, the new language. These small, new elements are then reconstructed into entities that represent (parts of) the theory, though in a new setting, a new language. This transformation I will now describe.

The translation is a three-step translation because it involves three elements: a theory, an intermediate conceptual framework and a technique. First, there is the theory (or parts of it) that is analysed. It is decomposed into smaller elements such as individual concepts, axioms, etc. These elements are still abstract. They refer to generic attributes that have no value yet. In parallel an experiment is designed. The experiment assigns value to the elements and attributes. E.g. a moral agent becomes Mr. Jones, age 67.

Both elements and values need to be translated into the elements from the intermediate conceptual framework[6]. A conceptual framework is required to a) guarantee consistency in experimenting and b) complete the translation to the technique. It defines all the conceptual elements and the logical apparatus that will be implemented using particular techniques. If the experimenter was to translate each of the elements of his theory into some concept or other separately, without making sure that each of these translations fits together with the others, an ill-fitted, overlapping, inconsistent set of concepts would result. A set that can perhaps be implemented, but that would almost certainly run into problems due to inconsistent instructions, contrary processing of information, etc. The intermediate conceptual framework really is the essential link between the theory on the one hand and the technique on the other.

The technique is a set of operational tools and artefacts that can be used to construct executable code. E.g. A computer programming language or a modelling tool. The subjects of experimentation must be able to demonstrate basic behaviour and functioning that is characteristic of the theory's subject matters. The technique contains the building blocks that are used to (re)construct the theory with other means. E.g. a human moral agent can act on certain information. If we want to experiment with an artificial agent we decompose him into several functions, such as an information processing and value assigning function, a decision function, etc. We make sure that he is internally consistent by casting these elements in terms of an intermediate conceptual framework. Each of these elements is then implemented using some technique like a computer programming language and computers, or robots with receptors, rotors, etc. Experimentation is really about decomposing and (re)composing: decomposing a theory and (re)composing it again in a different vocabulary, with different building blocks.

To make this transformation possible the experimenter must take several steps. Experimental philosophy, as depicted in figure 7, can be described in more detail as consisting of the following steps.

[6] I will discuss the intermediate conceptual framework extensively later in this chapter, and in the next section. For now just imagine the translation of, for example, a set of propositions about the change in scientific theories to coding statements in a programming language where the variable values are changed. These worlds are so much apart that a direct translation is impractical, if not impossible. To facilitate the translation the use of an intermediate conceptual framework, such as modal logic and game theories, is proposed. The concepts from such a framework are close enough to, for example, philosophy of science on the one hand and the Prolog programming language on the other hand, to make the translation practical.

1) Decomposing a selected philosophical theory into assumptions, premises, mechanism, predictions, etc.
2) that can be translated into a concrete experimental setting (a practical instance as opposed to abstract terms[7])
3) and translated into the elements of the intermediate conceptual framework.
4) The concrete experimental setting must be reflected in the intermediate conceptual framework.
5) The theory is implemented in the laboratory based on the requirements of the intermediate conceptual framework
6) reflecting the concrete experimental setting.
7) Experiments are conducted
8) and the results are translated back into the (restated) terms of the theory
9) that can be used to confirm, refine, reject, etc. the theory.

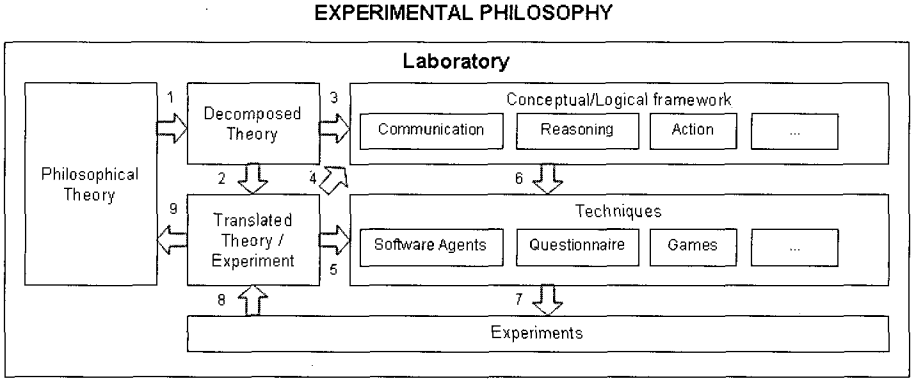


Figure 9: Steps of experimental philosophy

Referring back to figure 7, steps 1, 2 and 3 above constitute the step of abstracting and implementing. Steps 4 to 7 together make up the work of experimentation. Finally, steps 8 and 9 constitute the step of analysis and evaluation. The framework as just described, can be found underlying various work being done in experimental computational philosophy. Before launching into a detailed description of each of these steps, two paradigm cases will be discussed in the following two sections, demonstrating the aptness of the framework. These cases are Thagard's Conceptual Revolutions and Danielson's Artificial Morality.

[7] By concrete instance I mean the following. In the abstract a theory can say "It is wrong to lie." In experimentation this proposition cannot be used because it is someone or something (an artificial agent) that lies. A concrete instance thus refers to something like, 'Chris said he hadn't taken the apple where in fact he had.' Concrete instances can both refer to the 'real' world as to the virtual world of the laboratory.

They belong to the first group of researchers that not only foresaw the application of computers to philosophy, as for example Sloman did, but also put it into practice.
My descriptions below are not intended as a complete review of their work, nor as an assessment of its merits. I am interested in how they proceed in setting up and conducting their experiments. Their work will be presented just as far as necessary to provide the reader with enough background knowledge to appreciate what is being done, and to interpret it in the light of the methodology outlined above.
I will describe their work using the steps from the methodology. This ordering does not necessarily follow the original ordering in the work of Thagard and Danielson. Therefore, I will add references to the original section numbers as well. It will show that all the steps, as described in the framework, are taken by the authors, and no crucial steps are taken that are not in the methodology.
For convenience I have left out repeated references to the books I use in the following sections. For the section on Thagard my source is 'Conceptual Revolutions', (1992). In the section on Danielson chapter, section and page references without further source indication will refer to 'Artificial Morality: virtuous robots for virtual games', (1992). References to other works will include explicit identifications of the sources.

2.3 Thagard

Thagard is concerned with the philosophy of science. His focus in (Thagard, 1988, 1992, 1998) is on the development of scientific theories. Why, under what circumstances and how, do new theories replace old theories. Core in his theory are conceptual systems: systems of concepts connected through rules and relationships together with propositional structures in which they figure. Built around the propositional structures is the notion of explanatory coherence. This notion is key in Thagard's theory on theory development. It roughly equates to the coherence of a hypothesis with the other hypotheses and the data, and encompasses several principles that are extensively discussed and tested during experiments.

The structure of propositions can be analysed using a computer program ECHO. This program computes the strengths of the links between hypotheses, and between hypotheses and the data. It can decide between competing sets of hypotheses, that is, indicate which set explains most data (phenomena from the 'real world'). In addition, it incorporates notions of simplicity (the more explained with less the better), analogy, etc. His theory is substantiated with several examples and experimental runs of ECHO.

1. *Decomposing a selected philosophical theory into assumptions, premises, mechanism, predictions, etc.*

Chapter 1 provides the introduction to the theory. Thagard states his goal: to provide an explanation of scientific development, in particular major shifts. The notion of concept is central. Theories are understood as conceptual systems. Major shifts in scientific theories are conceptual revolutions. The theory is stated in six theses. They are important because they introduce the core elements of the theory:

- conceptual systems
- propositional systems
- kind-hierarchy
- part-hierarchy
- conceptual combination
- explanatory coherence
- abduction
- transition to new systems

Chapter 2 demarcates the theory by describing related approaches to the notion of concept. It describes what roles concepts play, p.22. This is important, as these provide a first description of the functional requirements for an implementation in experiments. Concepts are then defined as structures of relationships (with other concepts) and rules that govern the relationships. A conceptual system is defined as a network of concepts that are linked through part- and kind-hierarchies, and rules. There are five kinds of links. Thus, the static view of the theory of conceptual driven scientific theory development has been pictured and decomposed into small elements. In discussing the conceptual hierarchies, a reference is made to methods from the field of artificial intelligence (AI) to present knowledge structures. This is a prelude to later chapters, where the AI methods will be used as a intermediate conceptual framework in the translation to computer code.

The dynamic view is provided in chapter three on conceptual change. The notion of change is detailed (decomposed) in to nine degrees of change. Key in understanding change are the four mechanisms of discovery of new conceptual systems, and the replacement of old systems by the new. These mechanisms are described in detail[8]. "Developed by discovery" describes how the combination of old concepts can lead to new concepts whose features have value in solving

[8] I will here describe only the mechanisms of development and replacement by discovery. I leave the development and replacement by instruction out because they have less obvious links with the experiments.

particular problems. Combination alone is not enough, principles of salience, diagnosticity and casual reasoning also enter into the explanation. And finally, rule abduction, understood as the generalization from abducted hypotheses, and special heuristics, complete the picture of theory development by discovery. Thus, the mechanism is decomposed into smaller mechanisms and their relationships explained.

Replacement is described as the birth of new concepts that are linked in the old conceptual system. They do not fit the hierarchy of concepts satisfactorily. They have links with some concepts of the old system but not all. New concepts are introduced, and links are added and reorganised. Through competition among the rules, in terms of what they can explain, the focus gradually shifts from the old to the new system.

Chapters 2 and 3 present the theory and its elements. It is the step of decomposing the theory into smaller elements. Chapter 4 contains also theoretical elements, but has a much more operational focus.

The central notion, that ties concepts and change together. is 'explanatory coherence', discussed in chapter 4. It is defined as 1) a relation between two propositions, 2) a property of a whole set of related propositions, 3) a property of a single proposition within a whole set of propositions. More importantly, the notion of explanatory coherence is represented (decomposed) through seven principles, p.65. The principles "establish relations of coherence and make possible an assessment of the acceptability of propositions in an explanatory system.", p.65. To give an impression of the principles, I quote one of them, principle 2, "Explanation", p.66.

> *"If P1...Pm explain Q, then:*
>
> *(a) For each P_i in $P_1...P_m$, P_i and Q cohere*
> *(b) For each P_i and P_j in $P_1...P_m$, P_i and P_j cohere*
> *(c) In (a) and (b) the degree of coherence is inversely proportional to the number of propositions $P_1...P_m$."*

And with these principles described in section 4.1, the exposition and decomposition of Thagard's theory of explanatory coherence and conceptual change ends. From the concepts, their organization through links in hierarchies, and how they figure in propositions and the organization of propositional structures, we arrive

at their explanatory power. This is key in explaining the change from one scientific theory to another[9].

2. *that can be translated into a concrete experimental setting (a practical instance as opposed to abstract terms)*

The book is filled with examples from the history of science. The change in the 18th century from Stahl's phlogiston theory to the oxygen theory of Lavoissier is a prominent example that figures in chapters 3, 4 and 5[10]. Thagard analyses both theories and the discussion that took place. He shows the hierarchies of the concepts, section 3.2. Stahl subsumed calx and phlogiston under metal in the part-hierarchy (calx and phlogiston, he assumed, were contained in metal). Lavoissier on the other hand, proposed a part-hierarchy in which metal and oxygen were parts of calx. So it is not just a re-arrangement, but also a deletion of phlogiston and an addition of oxygen. In 4.3 the discussion of the replacement of the phlogiston theory by the oxygen theory is further detailed. The hypotheses and the evidence are listed in detail. These are used as input for ECHO (step 5).

3. *and translated into the elements of the intermediate conceptual framework.*

In section 4.2, the AI method that was mentioned in chapter 2, is introduced. It is connectionism[11]. Connectionism is, roughly stated, about networks of related nodes (tentatively equivalent to neurons). These nodes are linked through inhibitory and excitatory links. Thus, relations between nodes can be strengthened and weakened. The nodes represent hypotheses and data (evidence). The hypotheses can be linked to each other, one explaining or contradicting the other, or being analogous to the other. Data and hypotheses can also be linked, as hypotheses explain data, and data support hypotheses. As hypotheses cohere the link is strengthened. Likewise, if a hypothesis is supported by evidence the link is strengthened, and vice versa contradicting hypotheses weaken the link (inhibitory). The degrees in which links excite and inhibit hypotheses are parameters in the model. Two other parameters are the weight of evidence (not all evidence weighs the same), and the degree in which simplicity (if more hypotheses are needed to explain evidence the degree of excitation is lower) and analogy are weighed.

---

[9] Of course, this is a gross simplification. I leave many things out of his exposition and Thagard does not claim to have the theory that explains all about the transition from one scientific theory to the other. But for my purposes it sums up the core of the argumentation.

[10] Chapters 6 through 9 discuss other examples of scientific development. Though very interesting they are not really different from the phlogiston example. I will therefore leave them out of my discussion.

[11] Connectionism is what is nowadays known under the header of neural networks. I will stick to connectionism as Thagard uses it.

Connectionism functions as the intermediate conceptual framework. It cannot be said to be part of his theory of development of scientific theories. It is functional in restating or reformulating his theory. His theory has been decomposed into smaller parts (concepts, hypotheses, data, principles of coherence) that can be translated one-to-one to the constructs of connectionism, (nodes and links). Now Thagard has a translated theory that can relatively easily be implemented as a computer program. That this is a relatively trivial step, as I argued earlier, is evidenced by the fact that Thagard partly explains the connectionist framework using statements from the computer program that implements his theory (see step 6).

4. *The concrete experimental setting must be reflected in the intermediate conceptual framework.*

The hypotheses of Stahl and Lavoissier are translated to nodes and links in a connectionist network. E.g. Evidence/proposition 'In combustion, heat and light are given off', 'Combustible bodies contain phlogiston' (one of Stahl's hypotheses) and 'Pure air contains matter of fire and heat' (one of Lavoissier's hypotheses). These hypotheses are linked as explaining or contradictory hypotheses.

5. *The theory is implemented in the laboratory based on the requirements of the intermediate conceptual framework*

The translation of the experiment can be made only if the technique contains the constructs that represent the elements from the intermediate conceptual framework. The nodes and links are implemented in the ECHO program as functions. The functions are parameterized so they can be used to create concrete instances, like the phlogiston hypotheses in step 5.

6. *reflecting the concrete experimental setting*

In section 4.2 and 4.6 Thagard introduces the technique he will use to experiment. It is Common LISP, a computer programming language. Though he does not actually cite the computer code, he details part of it, writing down the pseudo code that resembles the actual code and detailing the algorithms that are implemented.

The functions implemented in LISP create a) units that represent the nodes, taking the name as input, and b) links between units together with a weight property. There are functions to create inhibitory and excitatory links, which are in fact connections of data units and propositional units.

7. *Experiments are conducted*

In section 4.3, Thagard runs his experiment on the phlogiston-oxygen replacement. An experimental run consists in this case of computing the network, that

is, the activation values of the nodes and weights of the links. Excitatory links increase the activation value, while inhibitory links do the opposite. The computation continues until the activation levels do not change significantly (the values increase and decrease asymptotically). The result is a set of activation values for each of the hypothesis. In the experimental runs, positive values indicate acceptability, negative values rejection. The values also indicate relative importance. Hypotheses with a higher value have a greater explanatory importance.

8. and the results are translated back into (restated) terms of the theory

The experiment shows that it is possible to model a connectionist network with nodes representing hypotheses (propositions containing concepts) and evidence. That such a network can be automated in a computer program. Executing that program provides results that confirm ideas about the correctness and explanatory power of the hypotheses.

The example from Lavoissier also shows that the combination of the theories of explanatory coherence and of conceptual change can be used to get a full picture of the dynamics of change. The dynamics are discussed in section 5.1.

In the experimental runs, positive values indicate acceptability, negative values rejection. The values also indicate relative importance. Hypotheses with a higher value have a greater explanatory importance. The experiment shows that Lavoissier's oxygen theory is better able to explain the evidence[12]. The values of the Lavoissier's hypotheses increase asymptotically to positive values, whereas the values of the Stahl hypotheses decrease asymptotically to negative values. Interestingly enough, one of the hypotheses of Stahl initially develops well, it explains particular data well. But as part of other hypotheses that do not do so well it looses strength.

9. that can be used to confirm, refine, reject, etc. the theory.

On the basis of the results in step 8, the theories of explanatory coherence, and of conceptual change are confirmed. It shows that the notion of explanatory coherence can be made operational and be implemented. It performs as predicted, and in coherence with our current understanding of historical cases. Moreover, it shows subtlety in that it can distinguish good hypotheses amidst a generally flawed theory. In chapters 6 through 9 other experiments are presented. The steps 4 through 9 are taken repeatedly. The outcomes are used to

[12] As Thagard uses the replacement of the phlogiston theory by the oxygen theory as an illustration it has not been crafted completely and contains a bias due to the fact that is based on an analysis of Lavoissier arguments.

illustrate the theory of conceptual change, but there are no theoretical modifications resulting from the experiments.

2.4 Danielson

Is it rational to be moral? That is the key question that concerns Danielson in his book Artificial Morality. At first glance rationality[13] and morality seem to be at odds. The ultimate test and proof that they are not incompatible is to have morally based behaviour, to compete in rationality contests and win. That is what Artificial Morality is about. Constructing a (con)test of rival strategies, of which some are based on moral principles, and other more or less selfish considerations. This contest is computer based, hence the 'artificial' in "Artificial Morality".

1. *Decomposing a selected philosophical theory into assumptions, premises, mechanism, predictions, etc.*

Danielson first describes the problem that he will focus on (chapter 1). Morality and rationality seem to be antagonistic. Danielson cites several examples in section 1.2, that illustrate that individually rational actions lead to situations in which all concerned are worse off. Driving in your car to work is individually pleasant except for the congestion. One individual's choice to carpool to work will not affect the traffic congestion in any noticeable way, nor will it affect the pollution, while it does limit your freedom to do as you like, and it increases the time it takes you to get to work. It is the well-known Prisoner's Dilemma. It is a compliance problem. A problem that has not been satisfactorily addressed by moral theories that focus too much on the common good and neglect the individual's interests, nor by rational choice theory, that neglects the instrumental usefulness of morality, p.10. In his search for a solution Danielson takes Gauthier's instrumental contractarianism[14] as a starting point (section 1.4). Though he finds Gauthier's attempt at solving the compliance problem inadequate, Danielson tries to formulate the answer within contractarianism. Danielson details the different approaches within contractarian theory. With Gauthier and Hobbes, and unlike Rawls (weak contractarianism), he seeks a justification outside morality (strong contractarianism). Like Gauthier, but unlike Hobbes, he chooses a strand of contractarianism that is not institutionalized (political con-

[13] Danielson uses the term rational as it is used in rational choice theory, as instrumental rationality.

[14] Contractarianism refers to the moral theory that bases the normative force of moral statements on the idea of a 'contract', a mutual agreement between the members of a community.

tractarianism) but internalized (moral contractarianism), that is, enforcement from within, and not from without.
Danielson seeks to give a fundamental justification of moral constraint. That is a justification from outside morality itself. In section 4.1 he further details the account of fundamental justification. He chooses instrumental rationality because it is non-moral and normative.

In chapter 2 the theoretical discussion really starts. First, Danielson details what a fundamental justification amounts to, "...a justification of a realm that does not appeal to any of the concepts of that realm.." p.19. This justification can be found in instrumental rationality. The compliance problem stands in the way of this justification, and must be solved by any theory that hopes to provide a fundamental justification. The compliance problem is decomposed into two aspects: 1) strategic failure, not being able to give a credible promise and thus missing the optimal solution, and 2), moral failure, moral agents cannot make credible threats against a would be defector and thus miss the optimal solution. In section 4.3.3, these problems are detailed and relabelled as an 'assurance problem'. To this problem is added a problem of an epistemological nature. How to detect players that are worthy of risking cooperation (with the risk of defection). This is the 'prediction problem'. In decomposing his theory several assumptions are made about parameters and mechanisms. It is assumed that it is possible for players to investigate the other players' decision function, and that the players can grant to, or withhold from each other the right to investigate their decision function. It is also assumed that the morally interesting problems are of the Prisoner's Dilemma type. Later, in chapter 9, this assumption is changed, and the games are modelled as the game of Chicken. Several experiments spring from the assumptions about the make-up of the population: how many players of each of the strategies are present in a population (see for example section 8.3). Some of the assumptions are made in the course of the experiment to solve an experimental, procedural problem. Sometimes, like with the mind reading assumption, the theoretical justification is given only afterwards. Other assumptions, that can parameterized in the experiment, like the population mixture, are made and altered in a truly experimental spirit: let's see what happens.

2. *that can be translated into a concrete experimental setting (a practical instance as opposed to abstract terms)*

Section 2.3 is used to illustrate the moral and strategic problems by means of games. Within game theory there are different types of games. The pure conflict games are characterized by the direct contrast of interests. These are morally not interesting. There is no mutually beneficial outcome that conflicts with individually optimal strategies. Mixed-motive games, on the contrary, have a mutually

beneficial outcome that is threatened by the individual temptation to break the (moral) rules. The experimental setting is still very generic, though the games have concrete, specific pay-off matrices. They reflect a structure rather than a concrete instance. They are illustrated by concrete examples. Step 2 and 3 are very much intertwined.
As an expression of the problems he has analysed in step 1, Danielson defines two strategies that represent the moral and the rational aspects of the problem. The (naive) morally motivated person will always seek the outcome that is mutually beneficial (co-operation). This works with other morally motivated players but leaves him open to exploitation by players that seek the individually beneficial outcome at the cost of others. These egoistically/rationally motivated players will always defect in seeking the outcome that is best for him. This works when confronted with a morally motivated person. But, when confronted with a like-minded, rationally motivated person the outcome is the worst one for both.

3. and translated into the elements of the intermediate conceptual framework.
Artificial morality is defined as the combination of game theory and methods from AI (section 1.5.2). AI appears later in the book to be predicate calculus implemented in Prolog (Programming in Logic ñ a language for computer programming). I will treat it as part of the technique, and refer to game theory as the intermediate conceptual framework.
Throughout chapters 1 to 4, elements and aspects of game theory are addressed (amongst others in sections 1.4, 2.3, 2.4, 4.1.3) and modelled: players as moral agents; games as situations in which players (agents) interact and that reflect the structure and the specifics of the situations; strategies are the actions; pay-off matrices reflect the interests of the players in the situation, how they value the different outcomes. Players embody principles implemented as decision procedures, p. 72. Danielson deviates from standard game theory in several important aspects. He uses interests, instead of individual preference orderings, as interpretation of the pay-off matrices. Interest is a less demanding concept than preference ordering. At the same time it allows for the existence of preferences beyond the current game. Danielson also abandons the assumption that all players are psychologically identical, and share common knowledge about themselves and the situation they are in.

I will not discuss the implications of these specific alterations. In general, however, it is important to notice them. It means that in fact two discussions are running. One about morality: are there morally motivated stratagems that are rational, and what do they look like. And the other about what is proper, and correct in game theory. Both are legitimate and interesting discussions. The danger is that they get mixed and arguments about one interferes with argu-

ments about the other. The reader has to be very much aware which part of the discussion he is reading about, and how the adjusted assumptions about game theory influence the moral arguments.

Danielson chooses one-shot, mixed-motive games as best suited for his experiments. One-shot games are games that are played only once. They are epistemically more demanding because there are no options to learn and retaliate in a next round. In iterated games, institutional solutions can arise. This choice of game reflects the theoretical stance taken in choosing for moral and not political contractarianism. The mixed-motive games bring forward the choice between the good (beneficial for all) and the rational (individually optimal). Danielson uses two forms of these games: extended and simultaneous games. In simultaneous games players execute their strategies simultaneous not knowing what the other player will do. In the extended version one player starts, and the other player follows, knowing what the first one did. The extended version clarifies and, at the same, simplifies the problems (section 4.4). These two variations in games constitute distinct experiments. In step 1 Danielson has decomposed the problem of morality and rationality into two problems, the assurance problem and the prediction (epistemic) problem. In a simultaneous game, both players have both problems. In the extended version, player 1 has the prediction problem while player 2 has the assurance problem. Though they seem identical in their structure and therefore need no separate analysis, later steps will prove this notion misguided, and thereby demonstrate the importance of the actual implementation of theories.

4. *The concrete experimental setting must be reflected in the intermediate conceptual framework.*

Danielson moves often from theory to experiment and back, designing a new strategy, implementing it, analysing the outcome, evaluating it in terms of the theory (is it morally acceptable and rationally justified in terms of pay-off, etc.). This is why I consider his work as paradigmatic, truly experimental. I will discuss two examples. The first experiment Danielson takes is with the basic strategies[15] that represent (in simplified form!) the antagonists. One strategy that is purely rationally motivated, and disregards considerations of mutual beneficence. This one is called unconditional defection. It is implemented as an individual player (see step 5). The next strategy is that of the moral agents that always decide on the basis of the common good. In his discussions Danielson has

[15] Each strategy is a proposition from a moral or rationality theory. They are an expression from game theory that restates propositions from the theory under examination. The conditional co-operator is the expression of the proposition that "one should strive for the common good but should not do so in a way that leaves one open to exploitation".

already pointed out the problems these strategies will encounter. The unconditional co-operator will be exploited by its non-moral opponents. The unconditional defector will never achieve the highest possible pay-off when playing with like-minded opponents. On a theoretical basis Danielson has designed a strategy of conditional co-operation. This strategy co-operates with other players that are willingly to co-operate, and defects with players that are not willing to co-operate. Thus it avoids exploitation and allows co-operation.

5. The theory is implemented in the laboratory based on the requirements of the intermediate conceptual framework

The concrete moral and rational strategies are implemented from section 4.3 onwards. Prolog is the computer language that is used. Prolog automates the reasoning with predicate calculus. Definitions of players implemented in Prolog can be seen as premises in an argument of which the outcome, the conclusion of the argument, is determined by the theorem prover of Prolog.
Following step 3, in which a player is defined as a decision function, a player is implemented as a decision function in Prolog. Strategies are labelled 'moves', and players are labelled according to their strategy (UD for unconditional defector, CC for conditional co-operator and UC for unconditional co-operator), the outcome of the decision function is either to co-operate or defect. Statements in Prolog consist of variables, constants and predicates. And, as any language, Prolog has its own syntax.

6. reflecting the concrete experimental setting

Each player has potentially two decisions to make: 1) what to do when he has to make the first move (remember it is an extended game), 2) how to respond if the other player started and his move is known. So a player is defined by two decision functions implemented in Prolog, like this example from p. 69.

```
% '%' begins a comment
% unconditional co-operator

m1(uc1, Other, c)
m2(uc1, Other, Anymove, c)
```

This Prolog code shows a predicate *m1* and *m2* that represent the decision function. In *m1* two constants figure, *uc1*, a reference to a concrete player, and the move that will be made, *c*, for cooperation. There is one variable *Other* that references the opponent.

Following the generic translation in step 5, the players are labelled according to their strategy. But now with a suffix added to individuate them. In implementing the conditional co-operator the need arises to determine how to find out whether the other player will co-operate or not. This reflects the epistemological problem that Danielson pointed out in step 1, and that now needs a concrete solution. The solution is the ability to execute the other player's decision function. It is coupled with the granting of permission to do so. In 4.4 Danielson discusses this same mechanism in the context of a simultaneous version of the game. This simultaneous version constitutes a different experiment.

7. Experiments are conducted

Conducting an experiment consists of executing the Prolog program. The Prolog code contains queries on the value of a variable. The theorem prover shows the outcome of the queries. A query to decide what a player should do is implemented as follows, p. 70

?- m1(uc1, uc2, Whatmove)
⇒ Whatmove = c

This query asks what the first move of an unconditional co-operator should be when meeting an unconditional co-operator. The answer is the value of variable *Whatmove* which is in this case to co-operate (*c*).

Running the experiments consists of coupling different players. Unconditional co-operators with unconditional co-operators, with conditional co-operators and with unconditional defectors in all possible combinations. Each run results in a particular pay-off for each individual player. The experiments show that a conditional co-operator does better against an unconditional defector, in terms of total pay-off, since he can avoid being exploited, while he can also fruitfully benefit from the co-operational attitude of both his fellow conditional co-operators. Something the unconditional defector is not capable of. Thus his scores are highest.

The second experiment is with the same mechanisms but now in simultaneous games. In this version his 'mind reading' solution does not work. It loops when an conditional co-operator meets a fellow conditional co-operator. Both execute the other's decision function. In this decision function, however, figures the other's decision function. So procedural problems bring the solution to a halt and force Danielson to design another mechanism. In the revised version of the strategy, the procedural problems are solved. The scores are now the same as in the extended games.

8. and the results are translated back into (restated) terms of the theory
The strategy of conditional co-operator outperforms the other strategies in terms of total pay-off (section 4.3.3), at least in the extended version. By this feat it is rationally superior to the rational, egoistic strategy of the unconditional defector. Since it does not exploit the naive unconditional co-operator, it is considered to be moral (or at least not immoral). The procedural problems in the simultaneous version require further attention (step 9). In the next run, after the problem has been fixed, the results show the conditional co-operator also outperforms in simultaneous games.

9. that can be used to confirm, refine, reject, etc. the theory.
It is possible to define strategies that are morally acceptable and rational. This is shown by the results of the conditional co-operator. To validate the strategy, theoretical adjustments must be made. One in the form of an assumption of transparency, a kind of intellectual property right that can be shared with others. The experiments also showed that the simultaneous games and the extended version are procedurally different, and that this difference matters. This, despite the structural similarity between the two versions. What works for the extended games does not work for the simultaneous game. The solution is found in the form of a meta-strategy in which the decision function itself is investigated, rather than its outcome. This, in effect, is a further alteration or refinement of his theory that follows from experiment.

The overall interpretation, at this stage, is that Danielson's claim that it is rational to be moral, is not impossible, to say the least. However, because several assumptions have been made that are possibly favourable to particular strategies caution is required: amongst others the assumption of transparency, the choice for the Prisoner's Dilemma type games and not the game of Chicken, and the make-up of the populations (section 4.4).
In the following rounds of experimentation (chapter 5 through 11) Danielson investigates the consequences of altering these assumptions, and of introducing more demanding and complicated mechanisms and strategies. The assumptions are altered and justified from a theoretical point of view, then modelled as a strategy in a game, implemented in Prolog and executed. The results are fed back in to the theory, thus closing the experimental cycle.

2.5 Some reflections

I have discussed two paradigmatic examples of experimental computational philosophy. The approaches are very different at first sight, but the underlying methods are similar. The differences are primarily differences in dynamics, not

in the steps that are being taken. It is fascinating to see theories coming to life in totally new vocabularies. I will now briefly compare their approaches and reflect on some aspects.

It is remarkable that Thagard introduces his intermediate conceptual framework and the technique he uses in just two sections (4.2 and 4.6). Neither can be assumed to be widely known, nor are they self-evident. The choice between threshold and sigmoid activation functions can have considerable impact on the performance of the network. This is an issue that is probably even more obscure to most readers than the general notion of connectionist networks. Similarly, it is likely that in 1992, when the book was published, there were a good many philosophers who had no experience with programming, or even with the use of computers. Let alone experience with Common LISP. Just as today there will be a good many philosophers who have no experience with the Java programming language or object-oriented design. This lack does not make them less legitimate readers. The author cannot be expected to recite the complete literature on the subject of the intermediate conceptual framework; but neither can the reader be expected to know all about the intermediate conceptual framework. The challenge for the experimenter is to bridge the gap. Part of this can be remedied by reading Thagard's Computational Philosophy of Science. Thagard discusses[16] some of the techniques and concepts he relies on in his later work, that I used as example.

Danielson spends more time introducing his intermediate conceptual framework and technique. The tests are extensively described and illustrated with software code. Though also in his case, it is likely that many readers are unfamiliar with computer programming in general, and Prolog in particular, the examples should not pose real difficulties for readers. This is because Prolog implements logic in a very direct, recognisable manner.

The dynamics of the two works are different as well. Danielson oscillates between theory and experiment much more often than Thagard. This is not a matter of better or worse, just a different way of operating. The difference might be due to two factors: documentation and research method. First, it might be that Thagard has just documented the final results where Danielson described the process of discovery as well. At least in part this seems to be the case when taking Thagard's previously published works into account. There is a close linkage between the cited books of Thagard. His theory develops over the course of these books, and culminates in the book discussed in this thesis.

Also their subject-matter is very different indeed. That a moral theory, focussed on interaction between agents, can be implemented does perhaps not surprise.

[16] In particular sections 1.3, 10.1, 10.2 and appendices 1C, 2 and 3

That a historical, knowledge oriented field as philosophy of science can also fruitfully utilize the computer and run experiments is a promising fact.

What the examples show, at the very least, is that by trying to implement a theory, both author and reader get a deeper and fuller understanding of what the theory is about. The implications of assumptions and mechanism become clearer. The fact that procedures do matter cannot be made clearer than in Danielson's case, where the experiment simply comes to a halt because procedures do matter, and have not been properly taken into account at a particular stage. In both examples, it becomes clear that craftsmanship is part of the job. A part that determines how far the researcher will get and how satisfactory the result be.

2.6 The methodology in detail

I will now describe each of the steps of the methodological framework in detail. They need not necessarily be taken sequentially. Steps 2, 3 and 4 can be taken simultaneously. Or rather iteratively at such small intervals that for practical purposes they can be regarded as simultaneous. The same goes for steps 5 and 6.

1) Decomposing a selected philosophical theory into assumptions, premises, mechanism and predictions

As we have seen above, experimentation requires translation to a different vocabulary. A theory as one entity cannot be translated. It consists of several elements such as premises, axioms, laws, mechanisms, etc. Each of these needs to be identified and named as such before a translation can be attempted.

2) that can be translated into a concrete experimental setting (a practical instance as opposed to abstract terms)

Experimentation is about particulars. Or, more precisely, about the application of general rules and laws to particular situations. Situations that characterize the application of the theory to concrete instances. They are simplified in that they are stripped of the irrelevant specifics. These situations, the experimental setting, must be designed. And next the theory must be applied to this situation. All mechanisms, axioms, etc. must be present and their impact identified. E.g. a theory about morality must be formulated such that it states that "in this or that particular situation, Elisa should (not) tell the truth about her brother stealing an apple from farmer John". The experimental setting is often an illustration used to clarify or test a theory because of the typicality of its application.

3) and translated into the elements of the intermediate conceptual framework
All elements from the decomposed theory and the experimental setting must be framed in terms of the intermediate conceptual framework. Now what is this framework? And why is it important? To understand, and fully grasp the importance of these steps, we must look a few steps ahead at the techniques. Before we can actually run an experiment it must be implemented in an executable environment. A biochemist has his tubes, a heat source etc.; the philosopher her computer programs. These programs must be written. Now there is an important difference between the biochemist and the philosopher. The biochemist has real samples of his subject matter whereas the philosopher has not. Though this does not make it necessarily easier or harder[17] it brings along different problems and questions. A computer program consists of code with statements. For each and every thing that is done or said some statements must be coded. From the perspective of philosophical theories a computer contains very little data or information that is directly relevant. It might have an operating system installed on it and programming languages, but still could not be said to contain any data structures or information that resemble notions like 'intention', 'good' and 'agency', that feature in philosophical theories. It does nothing and contains nothing until someone loads data and programs it to do something. I cannot stress this point enough because it implies that the experimenter has to have a view on everything. What does it mean to 'utter' a sentence[18], to hear and understand it? What does it mean to do something? When is something true? In theorizing about morality I can state that telling the truth is good. I can then focus on what 'good' means. When I want to do an experiment I have to program my agent so that it is capable of telling something . And when programming, I have to decide what 'telling' is. And no other agent will 'hear' anything until I program something that allows it to hear. And next, I have to indicate when, and under what circumstances, one agent hears what the other tells.

Having a view on all things is, of course, impossible. Nor will it always be necessary to the level of detail just sketched. But still, it will require an overwhelming amount of detailing and specifying. And that is where the intermediate conceptual framework comes in. An intermediate conceptual framework is a coherent set of rules and statements, of syntax and semantics that model particu-

[17] Experimentation with real samples is ridden with some difficult question like under which conditions the finding on small biological material may be extrapolated to conclusions about humans. Mass has substantial influence, for example, on how toxic substance affects biological organisms (a 100 kilogram human versus a 400 gram rat). And the findings cannot be extrapolated linearly.

[18] I use these words not literally. An agent on a computer can say something in the same a character in a novel says something. Which is expressed as a notation and not a sound wave.

lar aspects or forms of behaviour, of interaction, etc. Logical systems will be part of it, communicational and behavioural theories will be part of it, and much more. An intermediate conceptual framework provides ready-to-go packages with the concepts, mechanisms, rules, etc., that are required to set up an executable experiment. These are not part of the theory that is examined but nevertheless necessary.
Game theory is a good example of a theory that might play a role in the intermediate conceptual framework that is used to express another theory. In its standard form it specifies that there are agents, that their actions are expressed as strategies, that information is transparent, etc. Though it is not a complete set it will get the experimenter some way.

4) The concrete experimental setting must be reflected in the intermediate conceptual framework.

This step is the same as the preceding step. The only difference being what is translated. But once step 3 has been completed successfully this step will be prove to be trivial. Just as an example is an instance of the applied theory, an experiment is a parameterized instance of elements of the intermediate conceptual framework. E.g. if Step 3 involves translating 'to lie' to a statement with a particular truth value, Step 4 then would involve the translation of "my brother didn't steal that apple", which is a lie, to a tuple with a text string 'my brother didn't steal that apple', and boolean variable with value 'false'.

5) The theory is implemented in the laboratory based on the requirements of the intermediate conceptual framework

The elements from the intermediate conceptual framework are implemented using the technique chosen. This could be the work of the laboratory assistant. It requires mastery of the technique e.g. being able to program computer code. All elements from the intermediate conceptual framework must be restated but functionally equivalent. The constructs must be such that they can take on particular values as required by the experimental setting defined in Step 2. The ease with which this can be done determines the speed and flexibility of the experimenter. Though substantially not important (the usability does not determine the outcome) it is practically very important.

Core to the element of technique are:

- Resemblance or fit to theory: it must contain abstractions that can reflect and express the elements from the intermediate conceptual framework.
- Executability: the technique must be 'runnable', i.e. something observable must come out of the experiments.

- Control: the technique can be tuned in the various dimensions of the experiments (e.g. The number of agents, their attributes, the number of experiments, etc.).

The artificial constructs (the implementation of the theory) that are made using the technique must resemble the key elements of the theory as far as they are relevant for the experiment. Relevance is determined by whether they potentially influence the outcome or not. That is, by whether they function as an input or output variable, or as a parameter.
Self-evidently the experiment must be executed. Trivial as it may sound it has some practical implication that cannot just be put aside. Execution might require specific hardware and infrastructural provisions, etc.
Finally, the technique must enable the experimenter to configure and parameterize all relevant mechanisms, variables and parameters. In addition, she must be able to stop, (re)start the experiment and influence it mid-way. The freedom determines the extent to which the experiment can be tuned to the theses under investigation.

6) reflecting the concrete experimental setting
In order to run the experiment all particular elements in the experiment, defined in Step 2, must be implemented using some technique. This is, basically, configuring and parameterizing the elements from the intermediate conceptual framework that are defined and implemented in Step 5. The 'moral' software agents become Elisa and John, actions becomes uttering a sentence "My brother didn't steal..." implemented as a string variable in computer code. The translation itself is trivial though it can be laborious. Key to this step is choosing the values such that they reflect the theses that are being tested.

7) Experiments are conducted and
Now that all elements are constructed and parameterized according to the experimental setting, the experiments can be run. Execution of the experiment entails having the entities in the experiment act according to the rules laid down in the theory. Key to this step is registration of the configurations, parameters and results. It may, and hopefully will, sound self-evident. Tracing the runs and the outcomes is exciting work since it is here that the experimenter gets his first impressions about whether the theses are correct or not. Less exciting, though equally important, is the administrative aspect. Configurations and parameters and results must be registered in such a way that they can be analysed by the experimenter, and verified easily by third parties. Repetition of the results by feeding them again into the same techniques, as well as into different ones, is

essential. If the experiments cannot be repeated they are of little value in sustaining the theory.

8) the results are translated back into (restated) terms of the theory that can be used

The results from the experimental runs must be gathered, grouped and analysed. Analysis might involve statistical analysis as well as qualitative analysis. The results and analysis must be translated back to the theory. E.g. the fact that the average pay-off for strategy X is higher than for strategy Y in all situations Z, means that strategy X is Pareto efficient.

9) to confirm, refine, reject, etc. the theory.

The translated results can be used to evaluate the theory in question. In setting up the experiment, predictions and expectations may have been made about the outcome. The extent to which these are confirmed helps in evaluating the theory. The results might say something about the scope of the theory, hint at dependencies, etc. This step is not a part of the experimentation proper but the theorizing itself. New observations and evidence have come to light, and must be interpreted.

2.7 Intermediate conceptual framework

In detailing the methodology I have already pointed out the importance of the intermediate conceptual framework. This section examines the intermediate conceptual framework and its role in experimentation in more detail because its role is pivotal. There are two questions that are key. One, is (or should) the intermediate conceptual framework be a part of the theory under examination? Two, is its presence (always) required? The first question precedes the second. When the first is answered negatively the second is necessarily answered negatively. The first question is also a question of what the intermediate conceptual framework is about. Once that is clear, and if it is not a part of the examined theory, we have a basis for answering the second question.

What is, and what is not, part of a theory is neither arbitrary nor absolute. There are elements that are at the core of a theory and define it. And along a continuous scale the importance decreases. Where the border is drawn is very much a matter of pragmatics and custom.

One might be surprised that this 'old' issue of what a theory is turns up in experimental philosophy. In particular, no author has so far paid much or any attention to it. So, why does it come up? As I pointed out above, a computer is a machine with little semantics pre-loaded into it. Basically, it is a set of transistors

and electric currents to which we have attached the meaning of '0' and '1' (depending on the state of the transistor). The hardware is operated by machine code, low level programming, that connects with the hardware. Upon this layer rests the operating system that allows us to interface with the computer. This adds a layer of semantics to the machine, but hardly any that is relevant for philosophizing. This being the case, experimenters have to provide all the required meaning themselves in order to get started. Layer upon layer of increasingly complex constructs or artefacts have to be created. Not many authors seem to be aware of this fact. Many work more or less intuitively. So, why is this a problem? The experiments are running well. Moreover, what is so different with thought experiments that are written down in books? These books do not contain theories on all things that they might indirectly rely on. Referring to the quotes in the first section, this latter point seems to be a problem for traditional philosophy, rather than a reason why it will not be a problem for the experimental computationalist philosopher. The lack of rigour, sufficient detailing and completeness is lamented by the authors quoted. In chapter 5, I argue that a long-standing debate in philosophy between act and rule utilitarianism stems, in part, from an issue left unspecified in the theories concerned. Eliciting it, and putting a value to it, showed that the theories were not mutually exclusive, but instead located on different spots of the same spectrum.

Returning to the question of whether the intermediate conceptual framework is part of theory, I argue it is not. Making it part of the theory would overload the theory, and divert attention from the issues that really matter. The intermediate conceptual framework can be defined as all theoretical elements that are required to set up and run experiments, but that are not part of the theory proper. In this definition the content of the framework depends on both the extent of the theory and the richness of the technique. If, at some point in time, computers will be equipped standard with perception and reasoning capabilities there is no need to include those in the framework. Is the framework then not part of the technique? Again this is a matter of pragmatics. At least they are both part of SophoLab.

The intermediate conceptual framework is, as it name suggests, 'conceptual', that means that it is not executable. In addition, it is usually a part that is not configurable. Modal logic as part of the intermediate conceptual framework is not amendable during the experimentation. The concrete propositions formed during the experimentation are. Or more precisely, their executable equivalents are. This is why the intermediate conceptual framework is part of the laboratory as a separate element, and not as a part of the technique.

Another way to look at the intermediate conceptual framework is as common ground between theory on the one hand, and technique and tools used for the

experiments on the other hand. It recasts the theory into new constructs and derives from these the functional requirements for the technique. Its content will vary depending on both the richness of meaning of the technique and the abstraction level of the theory. Hence, the answer to the second question 'Is the presence of an intermediate conceptual framework always required?', is yes.

An excellent example of an intermediate conceptual framework is LORA (logic of rational agents), developed by Wooldridge (2000). This is an extensive framework broadly based on the work of Bratman, the BDI-model (belief-desire-intention model). It contains a modal and temporal logic apparatus to support the BDI-model plus extensions to support concepts of agents and actions. It allows agents to have beliefs, desires and intentions, to reason about them and to act upon them. It is well suited to model utilitarian research issues, though it is not complete.

The technique to be used would be computers with programs implementing the agents and the actions. There is elaborate software available, JACK (Agent Oriented Software Pty. Ltd), that implements the BDI-model supported with extensive facilities for reasoning patterns, inter-agent communications across computer networks, etc. It is written in Java and can be extended by the researcher with code for specific functionality.

The LORA/BDI-model constitutes the intermediate conceptual framework, the common ground between JACK and utilitarianism. LORA/BDI-model cannot be said to be part of utilitarianism. But it (or some other intermediate conceptual framework) is required to implement the theory as a computer program. Of course, the researcher could have made his own translation of utilitarianism to conceptual constructs that allow him to define what it means 'to tell a lie', 'to believe that action *X* will generate more utility than action *Y*', 'how to reason about actions in the future', etc. But it would be laborious, with the additional risk of inconsistencies between the different concepts in use. This risk is present because it is not the primary aim of the researcher, nor possibly her main capability. Arising as a by-product, it will probably not receive as much attention as it should.

Once this translation has been made, the researcher has to implement the concepts. Again this can be done from scratch or by using available software. This entails designing and implementing code that represents beliefs (for example as an enumerator holding string type elements that represent individual beliefs, and a logical variable holding the truth value of the belief). She would also have to design and implement methods to add, delete and modify beliefs. The researcher can do this herself, or use JACK or some other software package. This is of little consequence for the outcome of the experiment, though the effort to be invested can be enormous (but might be outweighed by greater control

over the program). It has some impact on issues such as verifiability and repeatability. Standard software packages cannot be modified easily, are better traceable and run on more varied computers systems.
Replacing the LORA/BDI-model could have considerable impact. In the current example, it could be replaced by classic game theory. Game theory and the LORA/BDI-model provide consistent and extensive frameworks to reason about agents and actions. However, they have marked differences. The LORA/BDI-model lacks notions of utility functions and pay-off matrices. The logical apparatus of game theory is not as sophisticated as LORA and it lacks sophisticated information processing (belief formation), and decision (desire and intention formation) functions. It will lead to different ways of modelling the issues. Though this does not necessarily lead to different outcomes, it is by no means certain that it will not. Hence, it is important to pay attention to the influence of the choice of intermediate conceptual framework. In section 2.7, I outline a way to express the impact of the intermediate conceptual framework.
The same applies a priori to the choice of technique. Given the choice for a intermediate conceptual framework the impact of the technique will most likely be smaller. Especially when the technique consists of computers and computer programs, because they can provide functionally identical systems through completely different implementations[19]. Techniques other than computer programs are currently rare but could have considerable impact. In that case, the same goes for the technique as for the intermediate conceptual framework.

2.8 Methodological issues

Pushing the application further towards actual use several questions come to mind. What, for example, is the influence of the intermediate conceptual framework and the technique on the outcome of the experiment? Does the recursiveness of experimental philosophy play a role? These questions will be addressed in section 2.8.1 on the neutrality of intermediate conceptual frameworks. Experimentation and its results must be shared with other researchers in order to benefit from them. In order to share, some standard should be adhered to. Also, colleagues in the field should be able, and allowed, to verify the experiments. These issues I will discuss in section 2.8.2 on standards of philosophical experimentation. In the following section (2.8.3) questions like "What is the relationship between theory and intermediate conceptual framework, and be-

[19] The use of procedural or object-oriented design and programming methods affect more the coding and efficiency of the computer code than they influence the behaviour of the code. Likewise the choice between programming languages Java, C++, LISP or Cobol has relatively little impact.

tween intermediate conceptual framework and technique in regard to translation?" and "To what level of detail must a theory be decomposed?" are discussed. Some other issues will have to await further research. How to design an experiment so that it optimally reflects the research issue at hand? What is the role of predictions in experiments? Do we always need them so that we can falsify a theory? There is also the question whether there is a set of standard elements that a theory has to broken down into, like axioms, propositions, etc. Or, does it not matter so much as long as the level of detail is sufficient for translation into experiments? Simply because it is too much to take on at once, these issues have been descoped. But also because these are secondary to the other issues. Only once the approach outlined in this thesis finds wider acceptance does it make sense to investigate these other issues.

2.8.1 Neutrality

Does the choice of intermediate conceptual framework influence the outcome of the experiment? The answer to this question has far reaching consequences because it determines both the validity of the outcome and the way in which to deal with the outcome and interpretations.

Say we have two competing philosophical theories Y and Z, and two intermediate conceptual frameworks, α and β, that can be used to implement the theories and conduct experiments. When trying to answer questions like 'Which of the theories Y and Z best explains a phenomenon q?', 'Which of the theories Y and Z best predicts the outcome of a particular situation q?' we implement the theories using a intermediate conceptual framework and model the situation of interest.

Does it matter whether we choose intermediate conceptual framework α or β? Possibly. I maintain that we cannot exclude a priori the possibility that a intermediate conceptual framework is biased towards a theory. The bias can derive from shared assumptions. This can arise accidentally or from the recursiveness I mentioned earlier in chapter 2. Because philosophy contributed to a theory outside the domain of philosophy (for example the use of Bratman's BDI-model in information engineering with multi-agent systems) and this theory is in turn used to investigate a moral theory. This moral theory and the BDI-model might or might not have some common roots that could explain a certain bias of the BDI-model towards the moral theory.

A bias can also arise because the fit between theory and intermediate conceptual framework will never be perfect, and, not necessarily the same for different intermediate conceptual frameworks. Due to a better fit the intermediate conceptual framework might be biased towards the theory in question.

We say a set of intermediate conceptual frameworks S = {α,..,β} is *strongly neutral* in regard of a theory *Y* over application domain q iff all intermediate conceptual frameworks yield the same results for *Y* when applied to *q*. A domain *q* is defined as a set of propositions that define a particular application domain, that is, the configuration of a generic situation that makes it a particular situation.

This means that modelling theory *Y* using *α* and *β*, and applying it to *q* yields the same outcome in both cases. The outcome of the experiments is thus independent of the intermediate conceptual framework used.

A trivial but illustrative example is using some form of temporal logic and fuzzy logic to model a particular situation. Under certain conditions both forms can be reduced to first order logic. For a domain that effectuates these conditions it is obvious that the results of using temporal and fuzzy logic will be the same. They can be said to be strongly neutral for that particular domain. But for other application domains it might be impossible to completely capture a theory by, for example, first-order logic. If two intermediate conceptual frameworks generate similar results / tendencies (e.g. same ordinal outcome) but differ absolutely, we call the set of intermediate conceptual frameworks *weakly neutral*. Finally, if a set of intermediate conceptual frameworks generates contradictory results the intermediate conceptual frameworks are *biased*.

What if intermediate conceptual framework favours *Y* over *Z*? In case of strong neutrality of *α* and *β* we can draw conclusions regarding the correctness or fitness of theories *Y* and *Z*. If *α* and *β* are both at least weakly neutral in regard to *Y* and *Z* (with the same outcome favouring *Y* over *Z*) we can make a weaker statement concerning the suitability of theories *Y* and *Z*. In this case and when *α* and *β* are biased in contrary directions we have some further research to do. Namely, why is it that *α* (*β*) favours *Y*(*Z*) over *Z*(*Y*)? Which underlying axioms, assumptions, etc. are shared by intermediate conceptual framework and theory? If we find shared assumptions that explain the bias, we have to make a statement regarding the validity of the assumptions, and thereby decide the question which theory to favour. In the above, I discussed the role of a intermediate conceptual framework. Exactly the same reasoning can be applied to the techniques used.

2.8.2 Standards of philosophical experimentation

Each experimenter, whichever domain he happens to be working in, is held to particular standards. These standards define what a correct experiment entails. These standards should be an integral part of the methodology.

Repeatability and verifiability are key issues in experimentation. Experiments must be repeatable. Not just by the original researchers, but also by other researchers. Repeatability means at least several runs with the same, and with

different parameter values. This again ensures that the results are not incidents . It also ensures a minimum level of stability of the experimental setting and the technique.

There are several forms in which repeatability can be framed.

a) Repeatable by the original experimenter with the same technique
b) Repeatable by the original experimenter with a different technique
c) Repeatable by the original experimenter with a different intermediate conceptual framework
d) Repeatable by other experimenters with the same technique
e) Repeatable by other experimenters with a different technique
f) Repeatable by other experimenters with a different intermediate conceptual framework

These degrees of repeatability are increasingly hard to achieve. Each next form changes an additional factor. Form (a) is fairly trivial. Each experimenter will repeat her experiments several times as part of the development of the experiments and in attempts to fine-tune the configurations. A slight variation is doing the experiment on different computers and/or different programming language. Each experimenter who has participated in competitions will be familiar with this slight variation. The difficulty seems to be more of an engineering kind. To program neatly, independent of the specific computer the program is run on, and even independent of the operating system, requires a structured, systematic approach that does not come naturally to all. Especially not to those whose primary domain of knowledge is not computer science. Repeating with a different technique, form (b), is laborious and fairly trivial. Programming languages can be made functionally identical. It is a different matter if the technique is not computer based, but say based on interviews, tasks performed, etc. In this case the difference does matter indeed. Form (c) seems to me to be the level that each experimenter should aim for, since it would prove beyond doubt that the results are not influenced by the intermediate conceptual framework used. It would also force the researcher to rethink and reformulate his theory not once but twice. Because different intermediate conceptual frameworks require different levels of abstraction it would be hard to obtain consistent results when the theory has serious flaws. Though it is not waterproof, it would go a long way in lending credibility to a theory.

Of actual repetitions in form (d) ñ (f) I have no knowledge. It would provide a good form of 'development and replacement by instruction', a way of learning about new theories by working with it, as Thagard defined it p. 58-60. The difficulties are the same as in the other forms, with the additional problem of someone other than the original experimenter having to carry out the experi-

ment. This requires that the other experimenters must have deep knowledge and understanding of the theory in question. But, if the aim is to convince other researchers to work with, and within, this new theory there is no way around it.

When it is hard to repeat an experiment (because of the scarce resources required) verifiability provides a useful substitute. Verifiability means that the steps taken during the experimentation must be traceable. All steps are written down in a protocol. This protocol shows what has been done how, when and under what circumstances. Outsiders can check the experiments without having to do them themselves. Currently there are no standards for such protocols in philosophy.

Transparency. Both repeatability and verifiability require that all modelling decisions are clear as well as all assumptions made, all parameter settings and configurations. This is a pre-condition for repeatability. Without them repeating is simply impossible. This information should always and standard be available and published together with the findings. In relation to verifiability they are a part of the protocol. The difficulty here is in being aware of the decisions and assumptions being made.

Openness. During all phases of the experiments, experimenters should allow observers to be present, as long, and in so far as, their presence will not obstruct the experiment. This is a form of transparency and verification during, instead of after, the experiment. It can be very useful and instructive because the experimenter can provide all kinds of information to the observer that is not formal and would not appear in the reports of the experiment. It would allow observers to question the experimenters in what they are doing and why. They will be better able to assess the quality of the experiment.

Since the practice of experimentation is still very new, it will not be a surprise to find that a few standards are adhered to already. My inclination is that most researchers go about their work in an off-hand, improvised, idiosyncratic way. This would be much in line with the original experimental practices developed in other sciences. As experimentation gains further momentum in philosophy we will have a need for standards though.

2.8.3 Functional equivalence

To what level of detail must a theory be decomposed to allow for experimentation? What is the appropriate abstraction level for the elements in the intermediate conceptual framework? Without the proper level of detail and abstraction experimentation is not possible. This is due to two factors. One, the behaviour of the higher level entities is embodied in lower level entities. A human (high level entity) can hear, but this ability is not derived from the complete human body but from the cooperation of several smaller organs (lower level entities), such as

the ear, and particular parts of the brain. Understanding of this ability is dependent on the modelling of these lower level entities. Two, experimentation is about translation, and translation is not possible at different levels of abstraction. I cannot translate the entity 'ear' from one language to the other if the other language only has a word for 'human body' and not for any of the organs it is composed of.

The level of detail required is relative to the research goals. It depends on the phenomena you want to investigate. If I want to investigate human functioning in general hearing is a part of that functioning and suffices as an abstraction level. On the other hand, if I want to investigate what it means to hear I need a more detailed level. The decomposition must be pushed further. This is the *minimally required* level of abstraction.

When (re)composing elements for experimental purposes one uses the building blocks from the technique. These are likely to have a level of detail that is too detailed for the elements of the intermediate conceptual framework. In the technique it is composition that is important. It constructs elements with higher abstraction levels. Consider the example of computers. Many computer languages consist of literals (invariant elements or constants) that constitute the basic data types (numeric literals such as integers, floating point numbers and booleans) plus a set of operations that can be performed on these literals (like appending a character to another character or adding numbers). These are indeed very detailed levels and form the *basic building blocks*. It would be meaningless and practically impossible to decompose a game theoretic intermediate conceptual framework to this level. So there is a level of *maximally possible* decomposition. With these building blocks entities have to be created that have at least the same minimal level of abstraction as is required by the maximal level decomposition possible. And preferably as high a level of abstraction as is in line with the minimally required level of abstraction.

Figure 10, illustrates the translation from theory to intermediate conceptual framework and the corresponding level of detail. For research purposes the theory must be decomposed to at least level (1). Further decomposition is meaningful and practically possible to level (2). The intermediate conceptual framework consists of building blocks with abstraction level (4). Further detailing is not possible. Using these building blocks, entities with increasing levels of abstraction can be created. I have added two variants of the maximum level of abstraction (a) and (b). What appears is bandwidth detailing what is appropriate and required for experimentation. In variant (a) the bandwidth is between (3a) and (2), and in variant (b) between (1) and (2).

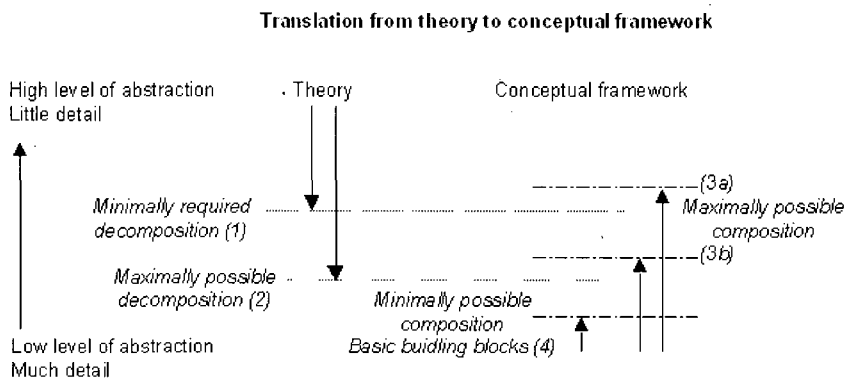


Figure 10: Translation and detailing

Within this bandwidth a level is chosen. The choice is pragmatically determined. How much effort does it take to further decompose a theory, or, to further compose elements from the intermediate conceptual framework. The only thing that matters is that the elements involved are functionally equivalent. Functional equivalence means that the elements perform functionally identically. A game in game theory *is,* of course, not the moral dilemma, but it has all the characteristics of the dilemma. One can ask where is the 'moral choice' and the experimenter should be able to point to an aspect of the game (choosing between strategy 1 and 2) that has no less and no more of the relevant properties. Materially they are implemented in a different way (just in the same way a description of moral dilemma is not the actual choice I have to make).

2.9 Evaluating theories

The translation from theory to executable computational model, as described in the previous section, requires the reformulation of the theory in the different vocabulary of the intermediate conceptual framework. Because this translation is the step towards execution it is much more rigorous. This in turn offers some interesting side-effects for the evaluation of theories. Mathematical and logical reformulations of theories allow for easier comparison of, for example, their explanatory power.

This opportunity to evaluate theories is a side-effect because this evaluation was not the intended outcome. The reformulation has as its intended purposes to bring the theory towards executability in order to be able to experiment with the theory. Theories can be evaluated on their empirical success and / or their problem solving capability. By this is, roughly speaking, meant the number of observations that corroborate or falsify the theory. In addition, there are theory

internal criteria, like consistency, that focus on the theory's make-up irrespective of its relation to the empirical data available.
A complete discussion of the various measures to evaluate a theory, leave alone their application to actual theories, is outside the scope of this thesis[20]. This section provides an indication which measures could possibly be applied.

- Provability

Provable in the theory means derivable from the axioms and primitive notions of the theory. The challenge for most philosophical theories is to define all axioms and primitive notions. Testing provability is not only useful from the perspective of having proved it, but it would also be helpful in determining some axioms or primitive notions are still missing. As such it would be useful in developing a theory.

- Consistency.

A theory is "consistent" if it never proves a contradiction. This is a basic sanity requirement for any theory. Most theories have never been formalized to such a level that this can actually be proven. Being able to do would constitute a big step forward.

- Completeness.

There are various definitions of completeness that give meaning to the general idea that a thing is complete if nothing has to be added to it. The definition depends on the application. Completeness in statistics is defined differently from completeness in order theory, for example. So the question of completeness depends on the intermediate conceptual framework used. Key questions for future research is to determine whether reformulating in terms of graph theory rather than mathematical logic has an impact on completeness.

- Decidability.

Decidability means there is an algorithm that decides in finite time whether a statement belongs to a set, or logical formula is valid in the system.
A logical system or theory is decidable if the set of all well-formed formulas valid in the system is decidable. That is, there exists an algorithm such that for every formula in the system the algorithm is capable of deciding in finitely many steps whether the formula is valid in the system or not.

[20] The evaluation of a theory is the subject matter of philosophy of science. In this thesis the criteria used to evaluate a theory is a given. It depends on the the particular field one is working in, and on the particular stance on takes in the various debates in philosophy of science. All that is intend here is show how the introduction of formal techniques through experimentation will facilitate the evaluation of a theory.

A decision procedure or algorithm is complete if, whenever the answer is "yes", the algorithm finds it correctly. It is sound if every time the algorithm answers "yes", it is the correct answer.
Practically speaking it means that once formalized we can determine for any statement whether it is a valid statement. This statement can be tested against data to check if theory and data coincide. More importantly, it will tell us whether the validity check can be done in finite time. This is a requirement for implementation. If validity cannot be verified in finite time the automation is impossible.

- Ockham's razor or theoretical parsimony

Explaining an observation in the least complex way possible. The expression of the economy of the explanation is dependent on the type of intermediate conceptual framework. It requires that two competing theories are reformulated in the same vocabulary, for example modal logic, in order to have comparable statements about the economy of the theories. The discussion of Thagard provides a nice illustration of the use of formalization to explain the transition of one theory as the running theory to another theory. The transition is driven by the explanatory power and efficiency of the theory.

- Complexity

The are several notions of what complexity is and how it is expressed. Of most interest in this context is the use of computational complexity. Computational complexity expresses either time or resources required to solve the problem. Given two competing theories the one that provides the solution fastest, given it is correct, is preferable.

Each of these criteria provide different ways to evaluate theories. Detailed discussion of how the criteria could be operationalized is outside the scope of this research. In particular the question is how the reformulation of a theory into game theory or modal logic is influencing the evaluation. Before this stage of sophisticated evaluation is reached the challenge is to come to a complete reformulation of a theory. This will proof a substantial challenge in its own right. It has to be addressed before the work on the evaluation criteria can be commenced.

2.10 Conclusion

The introduction of the computer into philosophy has introduced experimenting into philosophy. The use of computerized support has increased considerably over the last fifteen to twenty years. Behind the different research projects a shared approach can be found. I described a methodology that is descriptive of

many research projects involving experiments. It is descriptive of the work of Danielson and Thagard, two researchers with very different subject matters (morality and philosophy of science respectively). The dynamics differ as well. Danielson iterates through the experimental cycle from theory to experiment and back much more frequently than Thagard does. Nevertheless, their proceeding is basically the same. Key to the methodology is understanding that experimentation is about translating and reformulating a theory into another vocabulary. Direct translation from the theory to the technique that is used to run the experiments is not possible. As intermediary there is an intermediate conceptual framework that provides a coherent set of concepts that, in turn, can be matched to both the theory and the technique of experimentation. It is part of neither but can be linked closely.

3 Morality for artificial agents: implementing negative moral commands

Abstract

What does it take to provide artificial agents with moral reasoning capacity? This is one of the key questions of a research project trying to identify a possible approach to model and implement moral reasoning in software agents. In this chapter I will focus on the implementation of negative moral commands ("thou shall not..."). As building blocks the BDI model (Bratman, 1987) and the DEAL framework (Van den Hoven and Lokhorst, 2002) are used. For the implementation a software package for multi-agent system development, JACK, is used. The aim is to investigate some possibilities and limits of ethical agents. I contend that 1) through the attempt of implementation negative moral commands the understanding of the nature of these commands changes; 2) the logics of moral reasoning is suitable for software agents; 3) our moral theories are still too fuzzy to allow any artificial implementation of moral reasoning in a full sense. In particular, the moral epistemological aspects will prove a hard nut to crack.
Section 3.1 introduces the various kinds of ethical agents and defines the scope of this article. In section 3.2, the modelling and implementation components are introduced. Section 3.3 presents the implementation of the basic logic elements that will be used to construct moral behaviour. Section 3.4 addresses the implementation of the negative moral commands. The importance and impact of moral epistemology for artificial agents is discussed in section 3.5. In section 3.6 I draw some conclusions.[1]

3.1 Introduction

Various authors such as Floridi (presentation Uehiru/Carnegie conference 2005), Van den Hoven (presentation Uehiru/Carnegie conference 2005), Moor (2006) have proposed classifications of agents with increasing moral reasoning capacity and moral relevance. For the purpose of this article I will follow Moor's distinction of four categories of ethical agents: ethical impact agents, implicit

[1] I am grateful to Björn Jespersen, Gert-Jan Lokhorst, Jim Moor and Sabine Roesner for their comments on earlier drafts of this chapter or the paper that it is based on.

ethical agents, explicit ethical agents and full ethical agents. Ethical impact agents are all agents that have by their very nature and existence an ethical impact. The ethical aspect is not 'in' the agent but in the influence on its environment. Implicit ethical agents have moral considerations designed and built into them. But, they cannot be said to reason about the moral aspects. Their make-up is such that they simply cannot violate particular moral rules. Explicit ethical agents are agents that can make moral judgements, and provide some account of how they arrived at their judgements. Full ethical agents are agents that are engaged in making ethical judgements relating to complex, new situations with the ability to provide a plausible justification. A full ethical agent "lives a moral life" (Van den Hoven, private conversation). They have a free will, intentionality and consciousness.

Implementing moral reasoning in artificial agents is a very broad and complex topic. It is important to stress that the intention is not to construct anything that might lay a claim to getting close to human moral reasoning. Both the current technology and our understanding of the moral discourse are far too limited to even consider embarking on such a venture. This is not to say that it is impossible to create an artificial moral agent one day, that can compare to humans in its moral reasoning. But definitely not yet. My sentiment in this respect is the same as Wooldridge when working on his logic for artificial, rational agents.

> "Belief, desire and intention are in reality far too subtle, intricate and fuzzy to be captured completely in a logic [...] if such a theory was our goal, then the formalism would fail to satisfy it. However, the logic is emphatically *not* intended to serve as such a theory. Indeed, it seems that any theory which *did* fully capture all nuances of belief, desire and intention in humans would be of curiosity value only: it would in all likelihood be too complex and involved to be of much use for anything, let alone for building artificial agents." (Wooldridge, 2000, 91)

Hence, the focus in this chapter is on explicit ethical agents because they have some degree of autonomy, and provide a challenge to our understanding but are yet within the realm of the possible in the years to come.

3.2 Implementation

Implementation is done in three stages: modelling, design and coding. The first step is the modelling of the required behaviour. For this purpose DEAL (deontic epistemic action logic) is used in conjunction with the BDI (belief desire inten-

tion) model. These models consist of standard modal logic operators. They are used as specification language. This provides a language to capture requirements that is stricter than our everyday language, but more relaxed than the logic reasoning with axiomatizing and theorem proving.

Reasoning about what one knows or believes is captured by epistemic logic, which has two operators: B*i* (agent *i* believes that) and K*i* (agents *i* knows that). K*i*Φ states that agent *i* knows that Φ[2].
Action logic is a branch of modal logic. Its operator is STIT, "see tó it that".

(1) [*i* STIT: Φ] means agent '*i*' sees to it that 'Φ' is done or brought about

Predicate logic assigns predicates to actions and situations.

(2) G(Φ) means 'Φ' is 'G'

G can be interpreted as morally good and 'Φ' as a situation or an action that brings about some situation. Combining the above we can write

(3) G([*i* STIT Φ])

if an act is morally good - the act is good but the outcome might or might not be good.
Deontic logic has one basic operator,

(4) O(Φ) it is obligatory that Φ

Two other operators can be derived from this primitive operator:

(5) P(Φ)

it is permissible that Φ, or alternatively ¬O¬Φ, and

(6) F(Φ)

it is forbidden that Φ, or alternatively O¬Φ), (Van den Hoven and Lokhorst, 2002, 284).
Following Wooldridge's definitions[3]

(7) (Bel *i* Φ) means *i* believes Φ

[2] In the remainder I will use only the Belief operator as this fits with the choice for the BDI-model.

[3] Wooldridge designed an extensive and impressive logic for rational agents with many more components such as path connectives and quantifiers. For the current purposes the BDI operators suffice.

(8) (Int *i* Φ) means *i* intends Φ

(9) (Des *i* Φ) means *i* desires Φ

These elements can be combined to construct moral propositions. Consider the following proposition, which is for demonstration purposes only and not necessarily true or a desirable property of an artificial agent.

(10) $Bi(G(\Phi)) \rightarrow O([i\ STIT\ \Phi])$

meaning if *i* believes that 'Φ' is morally good than *i* should act in such as way that 'Φ' is brought about[4].

To create support for the above modelling components I will use the following implementation elements from the JACK development environment.

- Beliefsets - beliefs representing the epistemic dimension
- Events - goals and desires, for the goal-directed behaviour
- Actions, plans and reasoning methods ñ representing the intentions and action logic
- Agents - the container for the other elements
- Java programming language[5]

The deontic dimension is a complex dimension build up from the above elements.

Beliefs represent the agents view of its outer world. Beliefs are implemented as a first-order, relational model, called beliefset. Each beliefset has

- zero, one or more key fields[6] (all usual data types plus a logical member),
- one or more value fields,
- a set of queries[7].

The logical consistency of a beliefset is maintained automatically[8]. Beliefsets can be modelled using 'open world' and 'closed world' semantics. In closed world

[4] All the above provides still relatively simple moral propositions. Subsequent developments will increase the complexity, and problems like the problem of agglomeration, will surface. Mechanisms for reasoning about, and making decisions on conflicting obligations, are addressed in subsequent sections.

[5] In fact, the other elements are an abstraction layer on top of the Java programming language. Code extensions are not limited to these elements, but can be extended with specific, custom made Java code.

[6] A key field is a data field that is used for indexing, and can be used to find particular beliefsets.

[7] There are various kinds of queries: linear; indexed; complex – combining simple queries; function – developer coded, special queries. How they work is not important in this context. It serves to show that a fine granular set of mechanisms is available to unlock information.

semantics something is either true or false. The open world semantics allow something to be unknown. In the implementation the closed world beliefsets contain only tuples that are true. Tuples that are not stored are assumed false. In open world semantics both true and false tuples are stored. Tuples not stored are assumed unknown.

Desires and goals are what drives an agent in the JACK agent environment. They give an agent its goal-directed behaviour that allows it to reason about what it wants to achieve independently of how (which is done through actions / plans). It also allows for pro-active rather than reactive behaviour. Desires, as represented by `BDIGoalEvents`, are a special type of events. Events can be inter-agent or intra-agent. The former represent the usual interaction between entities, the exchange of information, requests and answers. The latter represents fine-grained internal reasoning processes. Events can be posted (for internal processing only), or sent (for inter agent communication) in various ways (polymorphy[9]). A BDI event can potentially be handled by multiple plans. When there are multiple applicable plans another event, the PlanChoice event, can be raised which is handled in turn by a meta-level plan (see plans below). Events can be posted or sent by agents from within plans, by external components (other programs), and by beliefsets

An agent has one or more plans at its disposal to achieve its goals. A plan is a sequence of atomic acts that an agent can take in response to an event. Committing to a plan, choosing a plan is like forming an intention. There are potentially several plans that can handle an event, and each plan can handle only one type of event. In order to determine which plan will handle an event (if any) there are two methods: `relevance()` and `context()`. The `relevance()` method determines which instances (all or some) of an event type can be handled. An event can carry various information which allows the `relevance()` method to determine whether or not to handle the event. From all relevant plans the `context()` method determines next which are applicable. The context method is a logical

[8] If a new fact is added that contradicts an existing one the old state will be knocked. To allow sophisticated reasoning (the agent may want to 'think' before really knocking an existing belief) events can be posted. Beliefsets can post events in case 1) new facts are to be added, 2) have been added, 3) the state of a belief changes (e.g. true to false) when new facts are added or removed due to negation or key constraints, 4) beliefs are removed. In response to the event the agent can decide whether to accept the change in the beliefset.

[9] Polymorphy refers to the possibility to have multiple implementations of the same basic function each with different inputs (signature in software engineering terms). The main benefit is that it allows for rich, fine granular behaviour.

expression that tries to bind the plan logical members[10]. For each binding a plan instance will be created. E.g. an agent might have a plan to help some other agent in need. But, it will only help agents from the same 'tribe', which is determined through the `relevance()` method in conjunction with information contained in the event message member. Next, it tries to bind a logical member AgentsInNeed against a beliefset containing all tribe agents, and an indication whether they are in need. For each of the bindings (tribe agent) a plan instance will be created. It might execute one plan, all plans till the first succeeds, or all plans.

A plan can have some meta information associated to it ñ accessible through `PlanInstanceInfo()`. This can be a ranking number that can be given a cardinal or ordinal interpretation. This information can be used to reason at a meta-level in case there are multiple, applicable plans. In that case, a special event, `PlanChoice` event, is raised. This event can be handled by a meta-level plan that facilitates reasoning about the various courses: the precedence of one over the other.

Explicit, meta-level reasoning is the finest granular reasoning facility. But plans also have a prominence, that is the order in which they appear in the agent's make up. If no other ordering information is provided, plans will be executed according to their prominence. Finer ordering can be achieved through precedence, providing a ranking to a plan which can be accessed through the `PlanInstanceInfo()` method.

When chosen for execution the body of the plan is executed. This is the core element of the plan that contains the detailed instructions (statements) of the plan. This is made up by the Java programming language, and extended with the JACK reasoning method statements. The reasoning method statements are special JACK agent language constructs that facilitate the control over reasoning, and specific agent behaviour. These statements are implemented as finite state machine[11].

[10] A member can be thought of as an attribute. A logical member is like normal data member, such as a string or an integer, but with the addition of following the rules of logic programming. Binding is the process of finding values for the members that match the logical conditions.

[11] A finite state machine is an execution model in which the execution of a step cannot be stopped but must be completed before anything else can be done.

3.3 Implementing negative moral commands

Using the equipment outlined in the preceding section I now discuss the way in which these concepts and ideas can be implemented in software[12]. In this section I will first show how the basic elements (operators) are constructed. Using these basic elements more complicated propositions (pairwise combinations of modal logic operators) will be created.

3.3.1. Belief

A belief that a state Φ is morally good can be implemented as a tuple describing that state in a beliefset of morally good states[13]. I will model two basic beliefsets: one representing the moral obligations, and one containing the specific actions or states with their deontic status[14]. I will model both under open world semantics. This requires the designer to be specific and as complete as possible. E.g. modelling obligations under closed world semantics will state as morally wrong everything the modeller forgot to specify as morally good. It also reflects better the fact the moral agents are not omniscient. Dealing with uncertainty is also an important aspect of moral behaviour.

1. Listing: beliefset MoralObligations

```
public beliefset MoralObligations extends OpenWorld {
      #key field String strObligationName
      #key field String strSphere
      #value field String strMoralProposition
      #value field String strText
      #value field String strType
      ....
}
```

[12] In this section the software components and the logical operators are used alternatingly. I will often when using one append the other in brackets to make clear what is being referred to.

[13] There is another option: implementing moral beliefs as a tuple describing that state plus its moral evaluation in a beliefset of all states. This option decreases the redundancy of information stored since all relevant aspects of a state are in one beliefset. When the moral dimension, and possible other dimensions, are stored in separate beliefsets the key fields need to be stored in all beliefsets increasing the redundancy. On the other hand, storing all aspects in one large beliefset increases the overhead of maintaining and querying that beliefset. It has a negative impact on the processing performance. I cannot say that one or the other might be closer to the way human cognition functions. The option I use offers greater clarity in representation.

[14] This status is derived at run time and not before.

The field `strObligationName` will contain a reference to the obligation[15]. The `strSphere` field is to recognize that obligations might have a restricted application domain, sphere. `strMoralProposition` contains the logical proposition representing the moral obligation, and `strText` its textual description. The `strType` is meant to be able to distinguish between states and actions. Beliefsets automatically have a boolean indicator signifying truth or falseness of the tuple.
The second beliefset contains tuples with statements about concrete instances of moral classes. If lying is forbidden, it will be a tuple in the beliefset `MoralObligations`. Saying "I'm a millionaire" is a proposition in the second beliefset. This second beliefset contains all states and/or actions for which it is relevant to know whether it is obligatory, permissible or forbidden. These are evaluated against the first beliefset.

2. Listing: beliefset MoralActEvaluation

```
public beliefset MoralActEvaluation extends OpenWorld {
      #key field String strActName
      #key field String strSphere
      #value field String strActProposition
      #value field String strText
      #value field String strType
      ....
}
```

The above sounds all well and simple. There are, however, some problematic aspects. First, how is an action classified? Or, how can an agent recognize it as being subsumed under particular class of moral obligations? E.g. how does an agent know that hitting someone without any cause is not permitted because it goes against the moral obligation 'not to hurt a fellow human being'. Give these questions some thought and it will become immediately apparent that it is far from trivial. It involves understanding the structure of an action or sentence, envisioning the direct and indirect consequences of that action, etc. Second, how does an agent know whether an act is morally relevant or significant? Me scratching my ear is a morally uninteresting action, but how do I know whether something is morally interesting/relevant? Or, how I can program an agent such that it knows what is of interest, and what is not? An option would be to train the agent using neural nets technology. This is possible, but results will at the current state of technology be very modest and the process long. We have to

[15] In this presentation I will skip several more technical details of the design that have no relevance for the moral aspects. Point in case is the reference. The actual implementation will have an unique ID for referencing purposes plus a name. In the presentation I leave out the unique ID as a field because it has no moral implications.

provide a fairly complete picture of what are morally salient attributes of acts and states. As these are to some extend situationally determined it will be clear that a complete moral classification will be impossible anywhere in the near future.
At the first step towards implementation it is clear that the logical modelling at the general level is not the problem. The problem arises due to either the absence of clear rules of what is (not) morally relevant in which situation, and, how to analyse actions such that they can be subsumed under moral rules. Our formal understanding of moral epistemology is too fuzzy to be implement for a general purpose agent. Humans can rely on their epistemic capabilities to be trained and learn to recognize situations that are morally relevant and subsume them under the appropriate moral rules. In the artificial context the epistemic capabilities are not yet advanced enough.

This does not mean that nothing can be done. What is required is that acts will have to be restricted and strongly typed (and hence classifiable). The basic structure of the reasoning remains but the classification, structuring and understanding of the acts will be exogenous, i.e. determined at design time. As time and cognitive science progress these design time decisions can be replaced by stronger epistemic capabilities that allow the agents to make these decisions at run-time. What does the short term solution look like? It means that for each act at least its object needs to be defined, the consequence of that act. The evaluation by that object of these consequences need to be known. This means extending the above beliefset with these attributes.

3. Listing: extended beliefset MoralActEvaluation

```
public beliefset MoralActEvaluation extends OpenWorld {
      #key field String strActName
      #key field String strSphere
      #value field String strActProposition
      #value field String strText
      #value field String strType
      #value field String strObject
      #value field String strConsequences
      #value field String strObjectEvaluation
      ....
}
```

The only thing that is now still missing is the actual evaluation. After the agent has properly classified the act it still needs to know whether the act is obligatory or not. To this end the `MoralObligations` beliefset needs to be extended with a query that evaluates an act against the moral obligations. After evaluating it returns an indication that it is obligatory, or not.

4. Listing: beliefset with function

```
public beliefset MoralActEvaluation extends OpenWorld {
      ....
      #indexed query getAct
            (String strAct, String strType, boolean bAct);
      #indexed query getConsequence
            (String strConsequence, String strType, boolean
      bConsequence);

      #complex query boolean getObligation (String strAct,
      String strConsequence, String strType){
            boolean bAct;
            boolean bConsequence;
            return getAct(String strAct, String strType,
      boolean bAct) && getConsequence(String strConse-
      quence, String strType, boolean bConsequence);
      }
      #function query getAllObligatoryActs (){
            ...
      }

}
```

What these queries do is first classify both the act and the consequences using the type indication. And then, based on the classification, query whether both act and consequence are morally obligatory, permissible, etc.

3.3.2. Desire

Contrary to beliefs, desires are easier to implement. A desire or goal is a special event, a `BDIGoalEvent`. It represents the goal-directed behaviour of the agent. The desire to behave morally can be expressed at various levels of detail and complexity. In its simplest form it would look as follows.

5. Listing: event as trigger for moral behaviour

```
public event BehaveMorally extends Event
{
}
```

This is admittedly a very crude version but would do the job all the same[16]. A desire is not effective until it is turned into an intention, and handled by a plan that can turn the desire into action. It can be tuned further to allow, for example,

[16] Depending on how the design is made an event can be kept active till there is a plan that can both process he event and successfully terminates. Thus, longer term, constituent behaviour can be created, rather than ad-hoc behaviour.

the tuning of the intensity of the desire; the domain (sphere) to which it applies; how, and by whom or what, it is instantiated. The below example shows the implementation of these extensions.

6. Listing: extended event as trigger for moral behaviour

```
public event BehaveMorally extends BDIGoalEvent
{
      int intensity; //integer denoting the intensity of
      the desire
      String strSphere; //application domain
      String strSource; //external source, e.g. father,
      mother,...

      #posted as ExternalMotivation (int intns, String
      sphr, String src)
      {
            intensity = intns;
            strSphere = sphr;
            strSource = src;
      }

      #posted as ReligousConviction (int intns, String
      sphr)
//there is no external source, conviction is internal
      {
            intensity = intns;
            strSphere = sphr;
      }
}
```

3.3.3. Intention

An agent has one or more plans. A plan is a sequence of atomic actions that an agent can take in response to an event. Committing to a plan, choosing a plan is like forming an intention. Reasoning about plans, i.e. about which intention to form, is done through meta-level plans. An obligation is a type of plan. It is a sequence of action which should be done. There is nothing fundamental to distinguish an obligation from a non-moral plan (from the engineering point of view!), as far as they are both a sequence of actions. Feeding someone poor and blowing my nose are both a sequence of actions. The distinction is added to the sequence in the meta information we attach to them, in the way we reason about them.

Whether an obligation is adhered to depends on how it is implemented. By tying it closely to an event, and giving it high precedence and prominence its execution can be forced. On the other hand it can be left to meta-level considerations. So an implementation can leave the abidance to the obligation open, and create some uncertainty. Particularly important here, is that at design time not all the

configurations need to be known (particular event information, precedence, etc. can be determined through configuration data which are read only at run time). A plan as a sequence of actions is a sequence of STIT operators. These are the atomic elements of the action logic.

3.3.4. Pairwise combinations

Above I discussed the basic building blocks. Evidently they make sense for an implementation only if combined to form complex constructs that express moral attitudes, reasoning, etc. As discussed above the operator O(), the existential and universal quantifier, and the moral attribute G() are implemented as tuples in beliefsets. As far as the structure and implementation are concerned (Bel i Φ), (Bel i G(Φ)), (Bel i E(Φ)), and (Bel i A(Φ)) are the same. Hence, I will only discuss the basic forms of intention, belief and desire. The deontic aspect can be added without loss of syntactical validity. The basic operators can be combined to construct morally meaningful propositions that can be used to represent desirable properties of moral agents. The intention now is to demonstrate how pairwise combinations of operators can be implemented using the software constructs there were introduced above. Consider the following three pairs.

i. (Int i Φ) → (Bel i Φ)[17]
ii. (Des i Φ) → (Int i Φ)
iii. (Des i Φ) → (Bel i Φ)

The A → B proposition can be given two interpretations. In the first interpretation, B is a necessary condition for A, the "conditional" interpretation. For implementation purposes one can, for example, think of the context() and relevance() methods which act as conditions for plans to be relevant and applicable. The second interpretation is one in which A is a sufficient condition for B. I will stretch this interpretation by giving it a causal meaning, that is A causes B to happen. I will call this the "causal" interpretation[18]. This makes sense in the

[17] Please note that these pairs serve only the purpose of showing how operators can technically be connected. In these basic forms they might or might not make sense, and be or not be, a desirable feature of a rational software agent (and a moral one at that). Proposition i) states that if i intends , i believes will be. There are many reasons why i might intend something which does not come true (something more urgent happening after the intention was formed). So the refined form (Int i Φ) → (Bel i E(Φ)), where E is the existential path qualifier makes sense, but (Int i Φ) → (Bel i A(Φ)), where A is the universal path qualifier, does not make sense. As I am only interested in the technical aspects of the connection I will disregard the plausibility of the proposition at this stage.

[18] The notion of causality is complex. By this informal use I do not intend to take any stance in the discussions on the nature of causality. The logical use of conditionals does not imply causation. Here, I use the term 'causal' rather than 'conditional' to indicate that in the implementation causality is intended.

context of my approach in which, for example, a change in beliefs causes an event (desire) to be triggered. Can the above three pairs, using these two interpretations, be implemented in JACK?

Ad i) Committing to a plan (forming an intention) requires the context() method to succeed. A context statement can contain a logical proposition with reference to a beliefset. In order to succeed the logical variable(s) need to be bound to one or more tuples from that beliefset. Thus, a belief can be a necessary condition for a plan. Of course, the beliefs can be there without the intention being formed.

A plan can operate on beliefsets, adding, changing or removing tuples from a beliefset. So there can be a causal relationship between an intention (plan) and a belief. Proposition (i) can be implemented under both interpretations of the A → B proposition type.

Ad ii) A desire (event) can cause an intention to arise. This is straightforward pairing where the plan is applicable and relevant to the event. If the plan contains a context() method the desire and belief both appear in the antecedent. So there is a conditional relationship, in which the desire and belief can figure as necessary conditions.

As an event (desire) can be triggered from within a plan, the plan (intention) can act as a sufficient condition. Again the relationship is straightforward. Proposition (ii) can be implemented for both interpretations.

Ad iii) An event cannot operate on a belief. A belief is modified either via external mechanisms and sources, or via plans. Hence, the occurrence of an event can never be a sufficient condition for a change in beliefs.

Events (desires), on the other hand, are sent through an automatic mechanism for posting: `#posted when ( condition ).` The condition contains a reference to the agent's beliefsets. In this way the beliefset functions as a sufficient condition for the sending of the event (the instantiation of the desire).

The discussion above is summarized in table 1 below. The column 'Conditional' indicates that the antecedent cannot take place without the consequent, the latter is a necessary condition. The 'Causal' column indicates the cases in which the antecedent is a sufficient condition for the consequent.

Proposition	Conditional	Causal
i) (Int *i* Φ) → (Bel *i* Φ)	√	√
ii) (Des *i* Φ) → (Int *i* Φ)	√	√
iii) (Des *i* Φ) → (Bel *I* Φ√)	√	x

Table 1: Operator connections

Please note that the relationships in this table are possible relationships. That means that they can be constructed as described. In the model, in an application, there can, however, be overriding relationships which may cause another relationship not to hold true. In the case of conflicting obligations an additional proposition needs to be introduced to detail how to decide the conflict.

3.4 Complex propositions – negative moral commands

If *i* believes something to be morally obligatory he form the intention to bring that something about.

$$(11) \quad (\text{Bel } i\, O(\Phi)) \rightarrow (\text{Int } i\, \Phi) \text{ or } (\text{Bel } i\, O(\Phi)) \rightarrow (\text{Des } i\, \Phi) \rightarrow (\text{Int } i\, \Phi)$$

This is a core notion and I will use it as starting point to investigate the implementation of moral notions. Before proceeding there is one further distinction to be made. There are 'obligations to do something' and 'obligations not to do something'. Though this might seem trivial it will become clear that this distinction has substantial implications. The obligation to tell the truth, is not the same as the obligation not to lie.

Let us look at the moral obligation not to kill a fellow human being. How to implement the adherence to the command 'thou shall not kill' ? Rephrase this as killing someone is forbidden, $F\Phi$, or $O\neg\Phi$, where Φ = killing someone. Say we have an agent with the desire to be moral and to adhere to 'do not kill', is expressed in (12)

$$(12) \quad (\text{Bel } i\, O(\neg\Phi)) \rightarrow (\text{Des } i\, \neg\Phi) \rightarrow (\text{Int } i\, \neg\Phi)$$

Implementing this obligation looks as follows. The agent has a beliefset in which the various moral obligations are stored amongst which $\neg\Phi$. I model the beliefset under open world semantics which means that it either holds true, does not hold true or is unknown. This means that it states which moral obligations the agents adheres (not) to. Based on this beliefset it posts a `BDIGoalEvent` for which it seeks applicable plans that help the agent responding to the event.
Now the next question is how to implement obligations? There are two options. Option one, there is a plan Φ representing the obligation. This plan takes precedence, if and when required, over plans to do the contrary. Option two, preconditions are added to all plans determining when they are (not) permissible. In this way the obligation cannot be said to have one location, instead it is spread across various plans.

Option one. When a particular desire arises and a `BDIGoalEvent` is raised a set of applicable plans will be selected. Amongst the various plans that are

applicable is also plan . As moral obligation it can be given a higher ranking and thereby pre-empt the other courses of action, in particular the ones that would count as violation of the obligation. Does this mean that a plan is permissible if there are no obligations to do the contrary? This is in fact what the formula P , it is permissible that Φ, or $\neg O \neg \Phi$, says. This seems to me to be problematic. Is shooting someone permissible in the absence of a plan not to shoot someone? What plans does the software agent have 'not to kill'? The answer is none. There is no positive act, no plan to 'not shoot'. Or the set of plans is non-empty and mostly meaningless in the sense that me drinking coffee is not going to shoot anyone.

The problem stems from the fact that killing someone is not an act! This is contrary to the way it is usually treated, and the way we talk about it. One cannot define *the* act of killing. This might seem counter-intuitive because we all know what it is to kill someone. Or, do we? Try to program an agent to kill someone. The agent would not know what to do directly if told to 'kill someone'. And it is equally problematic, if not more so, to define what 'not killing' is in a way that can be programmed directly as an act for an agent to execute.

We have many acts that count as killing, e.g. shooting someone, strangling, etc. It is impossible to exhaustively list all acts that count as killing someone, or more precise, a list of all obligations, not to strangle, not to shoot, etc., etc. This shows that practically speaking from an implementation point of view $P\Phi$, it is permissible that Φ, $\neg O \neg \Phi$, is problematic. 'Thou shall not kill' is a very imprecise statement. It seems to suggest an act, but it is in fact, about a state. It means 'thou shall not bring about a state of someone not having a beating hart any longer' (or whatever counts medically as being dead). Killing is a reference to a class of acts that bring about the same state. It is a reference to the consequences of an act[19].

In the same vain consider lying. I can breath, I can draw a line, I can smile, but I cannot lie. I can tell you I did not steal the money, where in fact I did steal the money. That would be called lying. But I cannot say "Do (not) lie!" in the same

[19] The above argument focuses on negative moral commands. The reader might wonder whether the same problems might be relevant to positive moral commands, e.g. to save a life. The reason for not including positive moral commands is that there is an important asymmetry between positive and negative moral commands in relation to acts. Whereas for both the number of acts that count as adhering to is endless in the case of negative moral commands it is important that all are excluded. For positive moral commands it is not important to known them all. In whatever way a life is saved is unimportant as long as it is saved. But with killing we want to make to make that all hundred are identified and stopped rather than ninety-nine, because the one undetected renders are other prevention meaningless.

way as "Do (not) raise your hand!"[20]. Try to program an agent 'not to lie'. It does involve a reference to what I bring about (my description of a situation) that does not match what I believe to be the case. So again it is a class of acts with a description that contains a reference to the consequence of those acts. This discussion shows that some obligations cannot be defined as acts. For now I conclude that option one, executing plans in absence of an other, overriding plan that represents a moral obligation, is problematic. The discussion of the implementation shows a different light on moral obligations that is interesting and might open new perspectives.

Option two. Pre-conditions, in the form of the context() method, can be used to control the execution of plans. They can be added, for example, to all plans to kill someone which would then possibly, on evaluation, fail and cause the plan to be excluded from the set of plans up for consideration. These pre-conditions can regulate when a plan should be up for consideration. It will make a reference to the beliefset containing all obligations and possible exceptions, and decide if it counts as a violation.

This option does not require a plan that is hard to conceive as in option one. The drawback is that the reasoning about plans is delegated to a lower level. When considering situations in which the act would not count as a breach of an obligation this might prove a problem because the meta-level reasoning is excluded. It also requires particular knowledge at design time about the moral system in which the agents will be functioning and the plan deployed. A plan can be executed in various situations in response to various events (desires). What counts as a valid reason (desire), as valid act (plan) in what circumstances is determined by the moral system under which the whole is executed. Some will be invalidated by all systems, but many will not. On the one hand, it is desirable to delegate knowledge at the lowest possible level in the system. On the other hand, duplication and inflexibility should be avoided[21].

When the application domain is limited, to say, health care, or even further to sub-domains like hospitals, the number of value changes is limited over a longer time frame, and can be captured relatively easy. Step by step the application domain can be extended without overburdening the design capabilities.

[20] Perhaps I am not pushing this line of argumentation far enough yet. There are obviously many ways in which I can raise my hand. This only emphasises the point I am making: one has to push towards the lowest levels possible in constructing behaviour.

[21] Korienek and Uzgalis (2002) make a very compelling case for redundant degrees of freedom in systems as this increases the adaptability of a system tremendously. The argument is made for artificial life systems, deriving from the study of biological systems. I am convinced that a similar case can be made for software systems as a subset of artificial systems.

Above I concluded that moral obligations are actually statements about states of affairs that are brought about by particular acts (plans). Acts that are classified based on the outcomes they produce. Following this notion one can combine option one and two and construct plans that have a pre-condition with a reference to the state that the plan may bring about. This variable should be checked against a list of states that may or may not be brought about. All remaining plans for which the deontic status cannot be determined upfront can be dealt with through the various mechanisms of meta-level reasoning. This would be a complete model in the sense that it would catch all plans (intentions) to kill before hand, without having to have an exhaustive list before hand. Also, the formulation and implementation of states is relatively straightforward. It requires the loading of a list from the configuration file at run-time containing all undesirable outcomes.
The catch with this revised option is that it assumes strong epistemic abilities. Ideally it should be possible for each plan to assess its impact and consequences. At design time this is at best partially possible. Another part will be dependent on the circumstances that cannot be foreseen at design time. The better the epistemic capabilities the better this options functions. If they are absent this option automatically reverts to option one, because no plans can be excluded upfront. To improve the performance even further plans can be combined into larger combinations of plans that form a capability. The content of each individual plan is reduced, and the 'intelligence' is contained in interaction between these plans. At the higher level plan, that integrates the lower level plans, the software environment contains a strong function to determine under which conditions a particular goal will be achieved. This function, `@determine`, contains a logical condition and a `BDIGoalEvent`. It finds the conditions under which the goal event can succeed (if at all). This provides an equivalent of 'internal' reasoning before the act to envision the consequences of an act.

The first tentative conclusion is that the mechanisms for the implementation of negative moral commands are available but practical only in limited application contexts. Moral commands are imprecise. This is their strength, they have a wide domain of applicability. However, it is also their weakness: they are vague and open to much debate and interpretation. Negative moral rules are short-cuts for defining classes of acts whose outcomes are undesirable and can be ruled out upfront without further consideration. When the application domain is limited it is practically possible to define such classes, as will be demonstrated in chapter 4. All the mechanisms are available but require intense computing, and/or a better understanding of our moral reasoning.

This leaves a last topic for consideration. What is the role and impact of epistemology? As noted the epistemic requirements can be strong. The last section is left for some initial observations on the role epistemology in the context explicit ethical agents.

3.5 Epistemology

What has not been discussed above in detail is how an agent comes to belief something, nor where its desires arise from. It will be clear from the above discussions that epistemology plays an important role in morality. In the implementation there are broadly speaking five elements of epistemological nature: 1) knowing the general moral propositions (the commands, rules, etc.); 2) knowledge of the actual state (of affairs), and the intended states and acts; 3) the projections of the consequence (future states) resulting from particular acts; 4) classification of acts and states under moral rules; 5) perception of moral attributes.

The importance of perception and cognition can be easily recognised. When considering the perlocutionary use of words (what the speaker intends to do by uttering them) and the moral implications, the question is how does an artificial agent distinguish the perlocutionary force of an argument. If goodness is said to be a non-natural characteristic that is supervenient on the non-evaluative characteristics of a situation, then how does an artificial agent perceive or deduce it? If ones duty is discerned intuitively, how does an artificial agent discern? What exactly is it that the agent discerns? If an agent beliefs the consequences of some act to be harmful to someone else, and hence refrains from executing this act, how did it come to hold this belief.

These are all questions that point to the epistemic capabilities of an artificial agent. None of these are easy questions. Answering them is outside the scope of this chapter. But without moral epistemology there can be no fully functional artificial, moral agent, unless one is willing to discard moral theories that rely heavily on epistemic capabilities. One reason for doing so could be the impossibility of implementing these capabilities, and hence the unrealistic nature of these capabilities. At first glance this surely is very unsatisfactory. It might have some merit, but insufficient to make such a claim right away. For now we could, perhaps just have to, accept that only a limited set of moral philosophies can be supported in artificial agent environments. Or, that the application domain is

limited such that the various restrictions, plans, etc. are known at design time and can be strongly typed[22].
One other key question is whether artificial agents have a need for the same type of morality as humans. I think it is desirable to provide artificial agents with some kind of morality, or rule abidance capability (relevant in any complex situation which cannot be modelled completely by an exhaustive set of rules). It seems to me at the current state of development any artificial construct is still limited in its capabilities. It subsequently has no need for a complex moral reasoning capability as humans do, yet.

3.6 Conclusion

Using various forms of modal logic I demonstrated how basic moral propositions can be formulated that also have an equivalent in software coding. The combined versions of modal logic suffice to model many relevant elements from the normative ethical discourse. The propositions are relatively simple but reflect sound moral insights.
Through the attempt to implement negative moral commands particular problems came to light. In particular, I did demonstrate that the use of moral commands disguises the actual structures of these commands. They seem to be about actions whereas in fact they deal with outcomes of actions. The drive to implement requires a detailed and minute formulation of our theories. Hence, I conclude that the encoding of theories in software is a great help in supporting and improving moral theories. It showed that our current understanding of moral theories is too fuzzy to allow for full blown implementations of these theories. Wider focus on various moral theories will show which ones are implementable, and to what extend.
Complexity and the still early stage of moral engineering mean that research will have to be restricted to limited application domains. The reasoning capacity will also be restricted to that of explicit ethical agents who can reasoning about moral situations, and, can account for their actions and decisions to some extend. This in itself is an encouraging finding and step forward. It should be mentioned that a complete explicit ethical agent still has to be built. Nonetheless, I hope to have

[22] There is a parallel development in speech recognition software. In the early stage of the speech recognition technology the first applications of voice recognition were in specific, clearly demarcated domains like law and medicine. Because of the limited application domain ambiguities in the interpretation of a word could be ruled out upfront, because in the limited context it could only have one meaning. Only after the advancement of technology, amongst others faster processing of large amounts of data, could the application domain be widened.

shown that there are no fundamental barriers to doing so. The most challenging aspect will be the moral epistemology. Moral theories make strong epistemological claims that will prove a hard nut to crack for moral engineers.

4 Deontic epistemic action logic: privacy & autonomous software agents

An executable approach to modelling moral constraints in complex informational relationships

Abstract

In this chapter we present an executable approach to model interactions between agents that involve sensitive, privacy-related information. The approach is formal and based on deontic, epistemic and action logic. It is conceptually related to the Belief-Desire-Intention model of Bratman. Our approach uses the concept of sphere as developed by Walzer to capture the notion that information is provided mostly with restrictions regarding its application. We use software agent technology to create an executable approach. Our agents hold beliefs about the world, have goals and commitment to the goals. They have the capacity to reason about different courses of action, and communicate with one another. The main new ingredient of our approach is the idea to model information itself as an intentional agent whose main goal it is to preserve the integrity of the information and regulate its dissemination. We demonstrate our approach by applying it to an important process in the insurance industry: applying for a life insurance.

In this chapter we will: 1. describe the challenge organizational complexity poses in moral reasoning about informational relationships; 2. propose an executable approach, using software agents with reasoning capacities grounded in modal logic, in which moral constraints on informational relationships can be modelled and investigated; 3. describe the details of our approach, in which information itself is modelled as an intentional agent in its own right; 4. test and validate it by applying it to a concrete 'hard case' from the insurance industry; and 5. conclude that our approach upholds and offers potential for both research and practical application.

4.1 Problem description

Some of the most pressing ethical issues involving the handling of sensitive information are to be found in complex organizations in which many people are engaged in different roles dealing with distributed information. Each has his particular set of right, obligations, sources of information and misinformation,

and so forth. The whole complex of informational relationships and interests becomes very hard to oversee. As Van den Hoven and Lokhorst (2002, 287) note:

> "...manual reasoning quickly gets overwhelmed. How should one delegate responsibilities, safeguard the flow of sensitive information, protect privacy, and so on, in today's complex organizational environments? Reasoning about such issues may be trivial so long as one is looking at the level of the individual agents, but the totality may be of mind-boggling complexity."

Information pervades every corner of life. Life is unthinkable without the technologies that have been developed to deal with all the data that we produce. Companies, private citizens and governmental organizations all deal with the use, application and distribution of data, but they do so from different perspectives. When the multitudes of roles that are involved are also taken into account, the complexity is very daunting indeed. The complexity arises from the numbers and fragmentation. The challenge it poses to moral reasoning arises from the fact that we cannot just extrapolate our moral reasoning from the individual-to-individual level to the organizational level. We do not know for sure that our moral reasoning still applies there. Moreover, we have at least an intuitive notion that it might not be applied without modification. In large organizations we separate tasks and the obligations and rights associated with them, while we distribute subsets of information that were originally provided as wholes, governed by specific sets of conditions and moral restrictions that were not designed with a view to the later partitioning .

With the rising intensity and complexity of data exchange, the need for instruments to control the use of data, to ensure its proper use and to prevent misuse becomes more and more important. Legislative measures are being developed and implemented to this purpose. But given the scale (unimaginable numbers of data transaction are being carried out each second) and scope (many transactions cross borders) it is unlikely that this will suffice.

The challenge for both practitioners and academic researchers alike is to find tools that abstract from the overwhelming detail while they are at the same time sufficiently rich to reflect the enormous complexity. In this paper, we propose to bring together several threads of research from various fields in an attempt to provide an approach that is a) formal yet practicable, b) can deal with complexity, and c) is executable. This approach can be used by researchers to set up experiments and to investigate, for example, emergent behaviour in large organizations; it can also be followed by practitioners to set up environments in which

the handling of information is more secure than when it is only governed by paper-based rules.

4.2 Solution

In our paper we present an executable approach to model interactions between agents that involve sensitive, privacy-related information. The approach is formal, based on deontic, epistemic and action logic. It is conceptually related to the Belief-Desire-Intention model (BDI model) of Bratman (1989). We add to our approach the concept of a sphere as developed by Walzer (1983) to capture the notion that information is provided mostly with restrictions regarding its application. We use software agent technology to create our executable approach. This serves two purposes. One, it enables academic research on a scale that cannot be achieved through armchair philosophy. Simply because the numbers are too big. In addition, preparing a theory for implementation is real challenge because it forces one to think of all elements that have been subsumed under the ceteris paribus clause, have been forgotten, etc. Two, it provides venues to operational application outside the academic realm.

Our agents hold beliefs about the world, have goals and commitments, form intentions. They have the capacity to reason about different courses of action, and can communicate with one another across any number of network domains. The key element of our approach is the modelling of information itself as an (intentional) agent in its own right, whose main goal it is to preserve the integrity of the information and regulate its dissemination.

We use different forms of modal logic to formalize information relationships. These forms have been brought together in DEAL, deontic epistemic action logic (Van den Hoven and Lokhorst 2002). DEAL draws upon several, well-known and widely accepted modal logics. We extend it here with the notion of spheres, which captures the fact the information has not the same status in different situations. Information acquired in one situation cannot just be re-used or distributed to different situations. However sophisticated, such a logic framework alone, however, cannot deal with complexity and is not executable either. The key to dealing with both complexity and execution is the same: providing a computer-based implementation of the logic. Using the computer to execute the logic potentially provides us with a means to handle real-life complexity. If and when proven satisfactory the same technique can be used for implementation and execution in practice. All this is easier said than done, though.

We propose to use agent technology (Russell 2003, Wooldridge 2000, 2002) as paradigm for our executable framework. It provides concepts that fit nicely to the situations that we would like to investigate: individuals in a particular capac-

ity dealing with information, sharing it with other individuals who may or may not apply, re-use, distribute that information, and so forth. This solution is scalable and executable as it runs as software on computers, the very environments where information is produced and stored. The implementation is done using a particular software package for constructing software agents, JACK (AOS, 2004). In addition to being based on the BDI-model it provides a good fit with modalities of DEAL. The fit, however, is not complete, obligations not being an explicit part of JACK. The addition of obligations is not problematic since they can be seen as an extension to the BDI model (Broersen 2001, Dastani 2001).

4.3 Conceptual aspects

In addition to the above generic implementation aspects, we add three aspects of a conceptual nature. One, we maintain that in informational relationships all events that set moral reasoning and activities in motion can be categorised as being of two types: a) requests for information, and b) changes in data. Two, the meaning of information does not have the same status in different situations, and might not even be the same in different situations. And at the time it is provided, this is generally done with implicit restrictions regarding its use. Three, information can be modelled as an (intentional) agent in its own right.

4.3.1 Triggers

Nothing happens without some trigger. In the case of informational relationships there are two, and no more, categories of triggers: a) requests for information, and b) a change in data. The first occurs when someone needs particular information to achieve his goal. He will look for sources that can provide him with the information and request that information. This triggers a reasoning process about who might have access to which data for which purpose, and may eventually result in actually obtaining the information. The second trigger sets in motion a consideration about who has the right to be informed about this change (a change can also mean the instantiation of new data), and who has an obligation to do so. Possibly, a third event might be the discovery of a wrong belief held by someone. If there is an obligation to prevent falsehood from persisting or spreading this requires than action. This event can be seen as a change in data, namely my set of beliefs that someone holds only true beliefs. Thus it can be categorized as trigger of the second kind.

4.3.2 Spheres

Walzer defines a structure in which people can organize themselves freely and still achieve some sort of equality, to achieve equality and fairness without falling back to some sort of tyranny. The key to his approach is his division of society in spheres. Spheres can be defined according to need and custom. Within a sphere a particular good is distributed. People are free to organize this distribution, on the condition that this particular good cannot be used outside its sphere to gain domination in other spheres. For example, domination in the political sphere may not be used in the sphere of trade. Moreover, in each sphere symbols, words, acts have their own particular meaning, which can very well differ from the meanings they have in other spheres.
This very notion of sphere can be applied to the context of informational relationships. Information gained in a particular sphere cannot be used to gain advantage in another sphere, at least not without particular, explicit conditions. Information provided to a physician cannot be used to evaluate the health status with respect to insurance, at least not without explicit consent. Rights to information and obligations (not) to provide information are related to the sphere in which it originated and the sphere of its intended use.

4.3.3 Information as an intentional agent

One of the key elements in our approach is to model information itself as an agent. An agent with desires, intentions and beliefs about its environment that is able to act. Its main aims are to maintain the integrity of the data, to provide access to all who have a right to request access, and to inform all who it is obliged to inform. Along with its desires, goals, and so on, it has the capacity to act and interact with its environment. This is very different from the current approaches in which there are one or more owners of the information. He or she guards the information and is charged with the task to inform those who have a right to be informed, grant or deny access to those who ask access to the data. In short, information is always tied to a (human) agent, it is passive.

The structure of the data and the content and context of the data determine the dissemination of the data. There is a clear analogy to the genetic code, which contains both information and the means to replicate itself. Such an informational agent is by no means a trivial thing to accomplish. It requires a clear conceptual distinction between the information itself, and it replication. On the other hand, there is a direct and complete dependence of one on the other. From the technical point of view this approach also poses some challenges. For one thing, the data as such should not be separable from the mechanisms that determine the dissemination of the data. Another interesting challenge is to

create self-aware data. Since the data is no longer a passive but an active entity, it needs some degree of self-awareness.

Modelling information as an intentional agent has several benefits. First, it ensures a consistent implementation and execution of informational rights and obligations, since all information is modelled and executed according to the same 'master template'. Second, as the execution of the actions that ensure the fulfilment of obligations and the safeguarding of rights is delegated to an entity that is no longer ridden with the potential conflicts of interest that beset its originator, it is more likely that those rights and obligations are effectuated. A human agent has several interests to cater for, apart from guarding privacy and preserving data integrity. Thirdly, as information becomes a self-enacting agent the effort of maintaining rights and obligations becomes less. Once instantiated they take over many of the tasks of administration. As information becomes more distributed, administration by a (human) agent becomes more time-consuming and likely to be forgotten or impossible due to loss of physical access (think of network failures, distributed networks with different access rights, etc.).

An objection to this approach is that it is too mechanistic, that it does not allow for the subtleties that characterize human interaction. For example, sometimes it is better not to provide all information and to tell a white lie. There are two rejoinders to this objection. One, the ethicist is just not specific enough to detail the requirements. Two, being a bit more strict is not necessarily a bad thing morally speaking.

Although an automated approach might run into the problem of being too mechanistic, this is not a fundamental flaw, but rather the result of our inability to express these subtleties. Once we have expressed them, there is no reason why they cannot be implemented. However, our present knowledge of moral behaviour, values, etc. is just to coarse to allow the expression of the subtleties we think we have in moral reasoning. I would be inclined to think that we often just do not fully understand ourselves we are doing when acting in the moral realm. And many argument around exceptions, subtleties, etc. is driven by ex post justifications.

One can also reason that a lot of harm comes from lies for ones own (or rather someone else's) best that fail to serve their purpose and from actions that are downright malicious. So not all subtleties work out as intended, and might even abused. On the whole the balance, even if one is inclined to say some good might come from subtle deviations from the moral rules, the overall balance might be in favour of a somewhat stricter application of moral rules.

4.4 The implementation

The experimental setting has been constructed using JACK, a Java extension with development environment. It is based on the BDI model. It offers a natural, conceptual equivalent for DEAL[1]. In this section we show how the conceptual elements, such as obligations, intentions, actions etc. are implemented. Table 2 provides an overview of all elements. Each of them is discussed in detail in the subsequent sub-sections.

Following the BDI model, JACK has agents that hold beliefs and data (tuples in a BeliefSet), have intentions (Plans) that prescribe how to achieve a goal (BDIGoalEvent) and interact with other agents (via MessageEvents). A plan consists of steps (Atomic actions) an agent can take in an attempt to achieve his goal. When several options are available the agent can reason about which course of action to take (Meta-level plan).

Logic element	**Implementation**
Agent	*Agent*
Predicate, Data, Information	*Class members/ Object attributes*
Sphere	As in Predicate, with extension of *context()* and *relevance()* operators
Epistemic/Knows/Beliefs	*BeliefSet* with *closedWorld* and *openWorld* semantics
Trigger change in data, request for information	*Modfact()* operator plus sending of an event *RequestThat{}*

[1] In setting up the experimental environment we had several requirements. First of all it must be formal, having a precisely defined syntax and semantics, whose properties are well-known. The implementation has to support the modelling elements. It has to support reasoning at a high level of abstraction. This means it has to provide implementation of reasoning concepts on par with the modelling constructs (such as intentions, action logic, etc.) - in short it has to be conceptually suitable.
In addition, support for meta-level reasoning is required. It has not just to be executable, but executable across different computer(networks), i.e. it must support distributed computing. New reasoning methods, support for new roles using new agent types should be integrated without have to adjust major parts of the implementation, its setup must be modular. On top of that it must be scalable. Since the core of the problem we are investigating is complexity arising from the large scale of operations scalability is of utmost importance. Support for industry standards for programming and computing is desirable when it comes to implementing the approach in real-life situation.

Logic element	Implementation
Action - STIT ('see to it that' operator) Inform{}, RequestThat{}	*Plan* with *@achieve* and/or *@insist* and/or @send statements plus *BDIGoalEvent* intending to change *beliefSet* or to get an answer
Obligation Permission, Forbidden	*Plan* with action statements plus *context()* and *relevance()* operators
Desire	*BDIGoalEvent*
Intention	*Plan*

Table 2: Implementation of logic elements

4.4.1 Agent and Predicate/Data

Software agents serve as actors in the theory. An agent is an autonomous entity with some (basic) reasoning capacity, the ability to form intentions and interact with its environment. At the current stage of developments such an entity is still very far removed from anything like human agents. This is not a problem for our purpose. In our implementation an agent is the key element, a container of knowledge and plans, the generator of desires and instantiator of communication. The attributes of the agent are the predicates in logic. These are forms of basic data. Information is presented as a complex data structure, that is, a combination of data elements.

An information element is, for example, an application file for a life insurance. It consists of three subsets of data which are related to each other: personal data, insurance data and medical data. This might complicate moral reasoning considerably because reasoning about the application file requires taking into consideration different rights and obligations regarding the subsets that make up the application file, and that might have conflicting implications about what is (not) allowed, obligated, etc.

From a logical point of view, data are represented by predicates. The computational equivalent in JACK is an attribute of an object or a member of a class. Young(Andrew) in predicate logic reads as 'Andrew is Young', where Andrew is, for example, a human. This would be implemented as

7. Listing: Instantiation of an object

```
public Human(String name, String seniority) extends Agent
      // two slashes forward means comment in the code
      // the variable 'name' has the value "Andrew", 'sen-
      iority the value "Young"
    {
      String Seniority = seniority;
      super(name);
    }
```

This code would be executed in the experiment by calling this method to create a new instance of the species Human.

```
new Human(Andrew, Young);
```

In this example a particular 'human', an extension of the class[2] Agent, is created and assigned the name "Andrew" with additionally the seniority "Young". The main difference with predicate logic is that an attribute has an explicitly used label, whereas in predicate logic the label is implicit.

4.4.2 Sphere

The notion of 'sphere' is conceptualized as follows. Each data element has an attribute 'sphere' that is set when the data is instantiated and can, under certain conditions, be changed during the lifetime of the data. When access is requested to the data, the intended application domain accompanies the request. This then is checked against the 'sphere' attribute of the data element.

The actual modelling is in part generic and in part specific to the domain. As a general rule the data-subject always has right to access the data and the right to disseminate them. The right to change them, however, is already specific. For example, an agent's health information is maintained by a physician who is responsible for the correctness of the data. The patient is not allowed to change the information about his health. He is, however, free to tell about his health to whomever is interested in his health, which is something that the physician is not allowed to do. We distinguish three basic roles regarding informational relationships: data-subject, data-administrator, and stakeholder. With regard to the role of data-administrator there are some complex issues involving delegation of the role and the associated rights and obligations. For example, a physi-

[2] A class is a construct in object-oriented programming that represents a particular type of entity (or data type). Its instances are objects of that particular type, that are generated in the execution of the software code.

cian may have an assistant to execute several of her obligations. Such an assistant holds the rights and obligations by proxy. The delegation is only partial and creates some additional obligations on the part of the delegator. Each agent with a particular role has a sphere attached to it in which it operates. The sphere defines the domains in which the agent operates and the restrictions it has in applying the information it receives.

In the implementation of beliefsets, which are discussed in section 4.3.3, the beliefsets contain a standard additional field 'sphere' where the sphere in which the data have been provided, acquired, and so on, are defined[3]. There are also fields to store information about the structure of the data, i.e., who is the data-subject, the data-administrator, etc.

Data and roles come together in plans. Plans are executed by agent in a particular role and aimed at acquiring, providing and withholding information. The execution of a plan is subject to two conditions: relevance and context. Relevance determines in general whether the plan is suited to handle a particular type of events, for example, whether a physician can handle a request for information from a patient. The context next finds out whether execution in the particular setting is allowed, for example, whether Dr. Elby is the physician of patient Drostel. Both operators are logical propositions that are evaluated before executing a plan and return either true or false. The operators are implemented as methods in the plans that try to unify logical variables with the available knowledge. If the unification succeeds the methods return the logical value true. Both the relevance() and the context() operators must return true before a plan can be executed.

4.4.3 Beliefs

An agent holds beliefs about the world. Reasoning about what one knows or believes is captured by epistemic logic, which has two operators: Ba (agent a believes that) and Ka (agents a knows that) (Van den Hoven and Lokhorst, 2002, 284). KaA states that agents knows that A is the case.
In our implementation knowledge and beliefs are captured in one component: beliefsets. We make no distinction between knowing and believing. The distinction is gradual. And although it is relevant in real-life, implementing the distinction between knowledge and belief is beyond the scope of this chapter.

Beliefsets are modelled as first-order tuple-based relations. The beliefs can be true or false, stating that the agents believe the statements to be true or false. As a refinement, we have two options: a closed-world semantics and an open-world

[3] For the reader interested in programming, this is implemented using a class sl_beliefset that extends the beliefset class and contains additional fields and methods to handle the sphere operations.

semantics. In the first, every possible statement has a truth-value. All statements that are believed to be true are listed. All statements not in this list are, by definition, supposed to be false. In the open-world semantics, both the false and true statements are listed explicitly. Statements that are in neither list are assumed to be unknown. The beliefsets can be queried using pre-defined queries. Beliefsets are implemented as follows:

8. Listing: beliefset query on patient health

```
public beliefset PatientHealth extends ClosedWorld {
     #key field String patientName;
     #key field String patientID;
     #value field String liverCondition;
     #value field String heartCondition;
     #value field String lungCondition;
     indexed query get(String patient, String ID, logical
     string lungCondition);
     ...
}
```

This example is about the health status of a patient. The beliefset contains information about the patient (his name and ID), and the health status of his heart, lungs and liver. The key fields indicate what part of the information can be used to query the beliefset. In the example we can ask after the health status of the lungs by using patient name and ID[4].

4.4.4 Triggers

To model the two triggers that set moral reasoning regarding information in motion we need a) two sorts of actions, and b) a notion of a 'change in data'. Both actions are derived from the primitive operator of action logic. The 'change in data' is captured by a combination of predicate logic and temporal logic.

Ad a) Action logic is a branch of modal logic. Its operator is STIT, "see to it that". [a STIT: A] means agent 'a' sees to it that 'A' is done. In the context of informational relationships that STIT operator can be supplanted by two more specific, more expressive operators: Inform and RequestThat[5]. These operators can be reduced to primitive modal operators in DEAL.

[4] There are many forms of queries possible from very simple to complicated, nested queries. An exposition of the possibilities is beyond the scope of this article. It is however a very powerful instrument.

[5] See Wooldridge (2000) for a detailed presentation of these operators as they are defined in LORA (logic of rational agents). Note that these operators are not modal operators or predicates but complex action constructs.

(13) {Inform $i\ g\ \alpha\ \varphi$}, i informing group g of φ through action α

This is a specific expression of the more generic form of

(14) [i $STIT_{\alpha}$: A] where A $\stackrel{def}{=}$ $K g\varphi$ and informing is specific instance α of STIT[6]

A request to someone else is expressed as

(15) {RequestThat $i\ g\ \alpha\ \varphi$ }, i performing α in order to get g to intend φ

{Inform} is used to model actions of informing other people about something. The request to receive information is modelled using the {RequestThat} operator. E.g. a request for information ω is modelled as

(16) {RequestThat $i\ g\ \alpha\ \varphi$ } where $\varphi \stackrel{def}{=}$ {Inform $g\ i\ \beta\ \omega$}

In our programming environment these operators are by execution of plans that contains atomic actions that send events containing some information to other agents. This is illustrated by the code below. An agent of type Patient sends a particular event HealthInfo using plan ProvideInfo to do so. It sends a message to Insurer, another agent. The message contains an attribute 'Value' that contains the actual information, in the example the value 'Good'.

9. Listing: Information exchange on patient status

```
public agent Patient extends Agent {
      #handles event RequestInfo;
      #sends event HealthInfo;
      #uses plan NewFact;
      #uses plan MoreInfoRequired;
      #uses plan ProvideInfo;
      #private data Health myHealth();
         ....
try   {
      myHealth.add("heart","good");   //initiate values
      myHealth.add("lungs","good");
      myHealth.add("liver","bad");
}
catch (Exception e){}
      .....
      }
}
```

[6] The sub-script on the STIT operator indicates that is a particular type of STIT operator.

```
public plan ProvideInfo extends Plan {
      ....
      #reasoning method
      body() {
         HealthInfo.Value = getHealthStatus("lungs",
         healthStatus); //get health of lungs
               @send("Insurer" , HealthInfo); //sending to
                  insurer agent
      }
}
```

Ad b) Change involves the concept of time. Concepts of time are introduced through temporal logic[7]. In logic terms change means that a predicate holds true at some moment and not at the next. 't' indicates a time point, t_1 time point 1, $t_1 \ldots t_n$ time points 1 through n.
A change in data is defined as

(17) $\neg \text{GoodCondition}(\textit{Liver})_{t2}$

which can be expressed using the temporal path connective $O\psi$ where O means 'next', so $O\psi$ is true now if ψ is true next, Wooldridge (2000:57). A change is thus defined alternatively as

(18) $\neg\psi \wedge \psi$

The computational implementation of this notion is as follows. A beliefset (a dataset) posts an event when an attempt is made to change data, after data have been added to the beliefset or after they have been removed, where the removal can occur because of inconsistencies with the current data or because of key constraints.

10. Listing: change in data

```
public beliefset Health extends OpenWorld {
      #posts event HealthChange evHealthChange;
      #key field String bodyPart;
      #value field boolean healthStatus;
      #indexed query getHealthStatus(String bodyPart,
      logical boolean healthStatus);
      .....
      public void modfact (Tuple t, BeliefState is, Tuple
            knocked, Tuple negated)
      {
            postEvent(evHealthChange.Unhealthy());
      }
```

[7] We adhere to the notional standard and definitions given by Wooldridge (2000:136 ff)

```
}
```

This example shows how a change in the health status triggers automatically an event, evHealthChange. This event in turn can trigger the execution of particular actions such as informing interested parties about the change.

4.4.5 Obligations

Deontic logic has one basic operator, *O*, "it is obligatory thatî. Its argument is a sentence that says that an agent brings about a certain state of affairs. E.g. it is obligatory to stop for the red traffic light, means that each agent that finds himself in a situation of approaching a red traffic light has to perform the action of bringing his car to a stop. Two other operators can be derived from this primitive operator: *P* (*PA*, it is permissible that A, $\neg O\neg$A), *F* (*FA*, it is forbidden that A, $O\neg$A), Van den Van den Hoven and Lokhorst (2002, 284).

In the context of informational relationships we have argued that there are two triggers that define all relevant instantiators of morally relevant behaviour: 1) a change in data and 2) a request for information. These trigger obligations, permissible or forbidden actions, e.g. informing an agent on the new information. Obligations are modelled as conditional modalities (conditional on the triggers) that lead to an Inform{} action. From the implementation point of view obligations are plans of action (an obligation *to do something*) that are triggered by an event. The events in turn are triggered either by a change in data or by an agent who wants to be informed. The execution of a plan is conditional on truth evaluation of two propositions: context() and relevance(). By constructing these propositions in such a way that they are true in all required situations, with the result that the plan is executed, we have enforced the execution of obligation. This is illustrated as follows.

11. Listing: obligation execution

```
public plan InformMedical extends Plan {
     #handles event HealthChange evHC;
     context()
     {
          evHC.Value > 20;
     }
     #reasoning method
     body()
     {
//this is a particular method that is called and informs
          InformPatient()
     }
}
```

In this example the obligation to inform a patient when his health deteriorates, the blood sugar is increasing, is implemented. A change in data triggers an event

(HealthChange) that is handled by a plan (InformMedical) if the value (evHC.Value) is below a certain threshold. The actual action of informing is defined in the reasoning method. Likewise, modalities such as 'permission' and 'forbidden', which are definable in terms of obligation, can be implemented.

In the above set-up a software agent cannot chose to ignore the obligation. This is a very limited implementation, which is perhaps not very interesting from a moral point of view. The inability to cheat takes away some of the most interesting questions. The up-side is that this mechanism provides a means "to protect privacy through technology rather than legislation". Our implementation has been extended to include different, conflicting interests. Meta-level reasoning is required to solve these. This brings back the full range of morally interesting dilemmas.

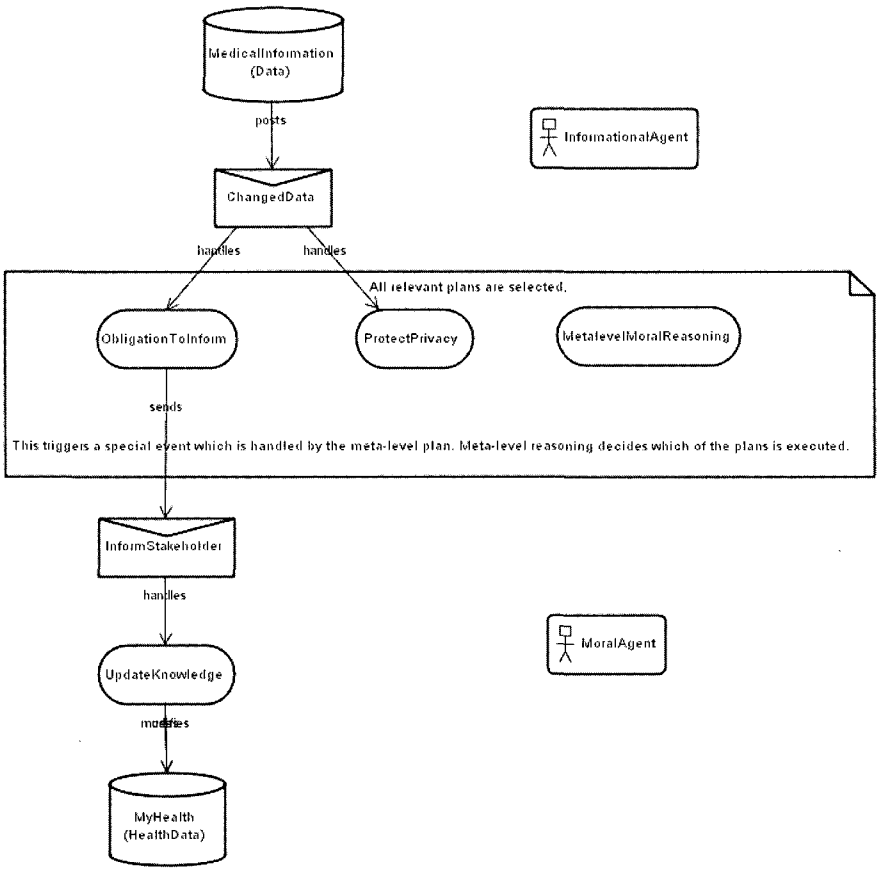


Figure 11: Obligation to inform on data change

We implement several plans that are all able to deal with a specific event, as illustrated in figure 11. Each plan represents a specific moral obligation or a chance to achieve a particular goal, and so on. Determining which plan takes precedence is a meta-level reasoning process. For this purpose a specific type of plans is implemented which collects all relevant plans and reasons about which plan to execute, or in BDI terms, which desire prevails and what intention is formed.

At the meta-level three different mechanisms are available to decide which one of the conflicting desires takes the upper-hand: prominence, precedence, and explicit reasoning. Prominence is defined by the order in which the plans are available to the agent: the first one is executed, and if it fails the next, etc. In precedence, plans are explicitly given a ranking that decides which one is chosen. These two mechanisms provide the equivalent of what Pollock (1995) calls the Q&I modules, the quick and inflexible modules that have certain rules quasi hard-wired into our system. Just as we do not need explicit reasoning in order to catch a ball that is thrown at us, most of us do not require any thinking to know that torturing a helpless animal is morally wrong. At times an explicit consideration of conflicting interests is required that take into account all current knowledge of the specific situation and weighs the (dis)advantages of each option. To this purposes meta-level plans (for example the MetaLevelReasoning plan in figure 11) are constructed that consist of atomic actions (e.g. getting additional information, using first-order and modal logic propositions)[8].

The way obligations are treated might seem different from the usual way of treating moral obligations. This can be explained by the fact the obligations and norms in the BDI context can be seen rather as possible extensions of goals than as fundamental constituting elements (Dastani 2001:8) . This does not, however, diminish the functional equivalence of the software implementation proposed. With the addition of the meta-level reasoning capability, the full moral reasoning spectrum can be supported.

4.4.6 Role-rights matrix

Rights and obligations need to be defined, even though their origin is not discussed in this chapter. They will be assumed according to need and custom. We will use a 'role-rights' matrix. It matches roles with the rights and obligations they have.

[8] In this one particular example the meta-level reasoning plan is empty. The purpose in this context is to decide on basis of prominence and precedence.

Relating to data XYZ (health related data)	Right to provide data (1)	Right to consent/ delegate right (2)	Right to use data (3)	Obl. to tell the truth (4)	Obl. to protect patient privacy (5)	Right to decide on acceptance (6)
Client/Data-Subject (R1)	Y	Y	Y	Y	N	N
Administrative personnel (R2)	N	N	N	Y	N	Y
Medical underwriter (R3)	N	N	Y	Y	Y	N
Insurer physician (R4)	N	Y	Y	Y	Y	N
Attending physician (R5)	Y	Y	Y	Y	Y	N

Table 3: Role-right matrix

If data XYZ is about the medical data of the client the tables specifies that, for example, in addition to the client himself, a physician employed by the insurer and the attending physician are allowed reading access to the data. Only the client can grant certain rights/permissions to other roles; he is not allowed to change his medical record, but this is allowed for his attending physician; etc.
The actions defined in the columns are the specific instances of the STIT operator and the deontic modality. The operators can be subject to specific epistemic conditions. The obligation to inform someone about something implies the knowledge about that something. Further conditions are derived from the sphere in which the data are being used and have been provided.

4.4.7 Desire and intention

An agent has desires it aims to fulfil. The desires are the motivating element that brings an agent to action. Desires are varied, and can be conflicting. Norms can be seen as a particular form of desire, or at least conceptually on par with desires. Resolving the relative importance of the various desires the agent commits to realizing a desire. In doing so an agent forms an intention. It devises a plan to achieve a particular goal, i.e. satisfy a desire.

To implement these concepts we have a particular type of event: BDIGoalEvent. This event type represents a desire. Posting it mean the agent will try to find one or more plans that can handle the event. This means finding the plans that

have the potential to achieve the goal. Selecting a plan to handle the BDIGoalEvent means forming an intention. BDIGoalEvent are special in the sense that in case there are several ways (plans) to fulfil the desire (achieve the goal) they can trigger another event that sets off a meta-level reasoning process.

4.4.8 Illustration

With the above equipment we can logically model informational relationships. To illustrate the expressive power of DEAL we now discuss briefly some examples (the numbers refer to the numbers in the role-rights-matrix in the preceding paragraph).

(1) 'everyone (x) has the right to provide (α) information he has about himself (A(x)) to anyone (y)'

$\forall x, \forall y, (P([x\ STIT_\alpha: KyA(x)]))$ or alternatively

$\forall x, \forall y, (P(Inform\{x, y, \alpha, A(x)\}))$

'every physician (y) must inform his patient (x) of any fact (A) about patient' or 'every patient has a right to know any fact about his health'

$\forall x, \forall y, (Physician(y, x) \rightarrow O [y\ STIT_\alpha: KxA(x)])$

or alternatively, 'physician (y) must inform patient (x) of any fact (A) about patient by phoning/telling/writing (T) the patient'

$\forall x, \forall y, (Physician(y, x) \rightarrow O (Inform\{y, x, T, A(x)\}))$

(4) 'no physician (y) is allowed to provide health information (A) about a patient (x) without consent from the patient (x) to anyone (z)'

$\forall x, \forall y, \forall z ((Physician(x,y) \wedge \neg\ Consent\ x) \rightarrow F([y\ STIT_\alpha: KzA(x)))$

(3) & (6) 'x has to the right to do A': $P([x\ STIT_\alpha: A])$

(2) 'Granting permission: if x has the right to do A then x has the right to grant y the right to do A'

$(P([x\ STIT_\alpha: A]) \rightarrow P([x\ STIT_\beta: P([y\ STIT_\alpha: A])]))$

4.5 Insurance and privacy

To test and validate our approach we apply it to a concrete case. We implement the process of application for a life insurance. This involves modelling obligations, permissions and actions that can(not) be taken by the different actors depending on their role, i.e. the broker, the insurer, the applicant, his attending physician and the insurer's physician. Life-insurance poses the ideal setting for experimentation with and testing of models dealing with information. It is complex because there are many roles, it is relevant in real life, it deals with privacy-sensitive data, it has potential conflicts of interest, and it has a long time span, which makes it apt to change. It serves to show the relevance of the approach to practice.

4.5.1 The test case

The insurance industry is a particularly information-intensive industry. The information is, moreover, sensitive, and, at least in the case of life-insurance, bound to strict rules as to who has access to which information. In the processing and administration of policies a number of different people are involved. The applications and policies themselves contain different subsets of information, such as client data (e.g. address, marital status), insurance information (e.g. what policy, which coverage) and client health statements. This information is used in several, different settings: to decide on acceptance of the client's application, in generic analysis on risk profiles (e.g. which body-mass index has a higher risk profile regarding certain diseases and should therefore possibly have a higher premium attached?), in administrating the policy, etc.

We will now describe a concrete instance of sensitive information handling in the insurance industry. It captures the core elements, but abstracts away from many details. A (potential) client (R1)[9] applies for a life insurance. To decide whether to accept the application or not the insurance company requires particular information. It uses an application form that the client has to use to submit the required information (data). The application form consists of three separate sub-forms, one containing generic client data (D1), one containing information regarding the requested product (D2), and one containing medical information about the client (D3). The client provides this information for use in deciding on his application (S1), with the understanding that this information will be treated confidentially: the information will not be used outside the insurance company, and the use is also restricted within the insurance company. It can be used in anonymized form for policy-making purposes (S2). The client sends his personal

[9] R denotes a role, D data and S a sphere.

data and the requested product details to the new business application department of the insurer, where it is processed by an administrative employee (R2). The medical information is sent separately to the medical department, where it is processed by a medical underwriter (R3). The medical underwriter consults, if necessary, the insurer's physician (R4) to evaluate the client's medical history and current status in regard to the insurance company acceptance policy (D4). If the provided information is insufficient to come to a conclusion the underwriter informs the client that additional information is required from the attending physician (R5). If the client wants to proceed with the application he is required to give his written consent (D5) stating that the attending physician is granted permission to discuss the client's medical data as far as relevant for the application, and in reply to a concrete question that concerns a specific section(s) of the medical form/questionnaire (S3). When the consent is given and received, the insurer's physician writes a request to the attending physician asking him to provide information about the nature of a specific ailment/treatment (D6). The attending physician is bound to tell the truth and protect his patient's privacy. Now in this context several situations can arise in which interests conflict or unintentional errors are made.

1) The insurer asks the attending physician information about aspects that are not covered by the consent.
2) The attending physician sees that the information provided by the client/patient about other aspects than those covered in the consent is false.
3) The insurer or an individual employee sells data to a pharmaceutical firm for direct mailing or generic, postal code based marketing.
4) The applicant sends medical information in the wrong envelope, as a result of which it ends up with the administrative employee instead of the medical underwriter.
5) Marketers request information about policyholders for policy-making purposes.

This is only a small list of many things that pose potential conflicts of interests, decision points for actors. In the above rudimentary example, with six data elements, five roles and three spheres complexity is already considerable. If one consider the role-rights matrix, table 3 in section 4.4.6, which shows only a limited set of six actions, in addition to this, it is easy to understand how complexity grows exponentially. The potential for misinformation, mis-use, and so on, is enormous.

4.5.2 Results from the experiments

Setting up the experiment

Setting up the experimental setting meant designing and implementing the software agents. We distinguish two basic types: informational agents (information) and moral agents (representing human actors). The latter are further extended to represent the different actors such as physicians and applicants. Agents have plans at their disposal to inform other agents, to grant access rights to others, to request information, to withhold information, and so forth. These plans deal with events, use data that the agent has, and so on (see figure 12). These elements have been set up generically. They are parameterized so that for experimenting purposes plans, actions, and so on, can be changed without having to rebuild the basic constructs. Parameters are read via a configuration files at run-time. The advantage is that in this way the scale of the experiment can be increased without effort.

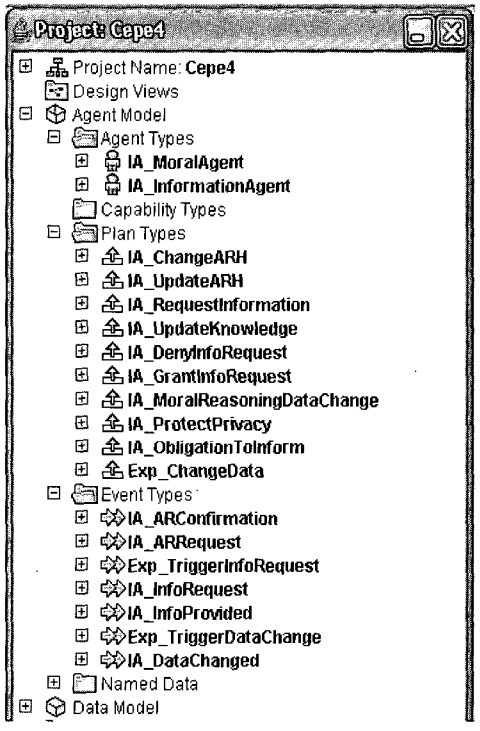


Figure 12: Software constructs in the experiment

Is it possible?

Running the experiments shows it is possible to capture all the moral concepts that are relevant to informational relationships in an automated environment. Actors, obligations, rights, acts, and of course, information itself, can all be modelled and implemented in executable code. In addition to this, the particularly challenging aspect of meta-level reasoning is captured using (meta-)plans. Software agents represent moral agents. They have several, different and sometimes conflicting courses of actions available to them. Meta-level reasoning facilitates the choice between these courses of action. Meta-level reasoning itself is facilitated at three different levels of sophistication.

Defining and implementing the agents, the plans and the events proved to be substantial work. A surprising finding was the relatively low level of complexity of the individual plans and events. The context()propositions usually contain one to four logical variables. The plans themselves are not very complex in a programming and logical sense. They send events, read data, update their knowledge base, and so on. The challenging part is in the chain of plans and events and the complex web they constitute. After adding several agents with their plans and events, the number of potential relationships soon became daunting. Keeping track of all the elements proved very hard though we had an automated environment available to assist us. We found that having parameterized plans and events helped in dealing with the complexity. Nonetheless, even with the technical support it proved hard to keep an overview. We believe that without the technical support we would soon have been lost, and unable to oversee the consequences of our modelling decisions. Interesting in this context is also that complexity of administrating the experiment decreases after the initial increase. Once all basic mechanisms are put in place, extending the experiment with new agents proves fairly easy.

Information as an intentional agent gives clearly more control over information spreading and availability. The price paid is an increased overhead in communication. Rather than mental or technical internal reasoning within a human's mind or a software agent's own classes, communication across domains is required. In technical terms this means longer processing times sending triggers and data across. These processes are also more CPU intensive. Although this not a technical problem on the present scale, it might become one when the scale is enlarged.

Two types of triggers for moral reasoning

One of the benefits that come from the implementation of a theory is that it requires a complete specification of the theory. There is no room for ceteris paribus clauses, nothing can be assumed given and nothing can be overlooked.

One of the challenges in implementing the informational relationships was the distinction between the two types of events that give rise to moral deliberation: change in data and request for information. Two questions arose: a) can these two different types be handled by the same reasoning", and b) do they have the same moral status?

In our approach both types can be handled by using the same techniques and elements. Functionally and technically they are identical. The difference is not in the execution (the act) itself, but in the conditions under which it is triggered. This showed itself when implementing the constructs required. Although the plans implementing the obligations have the same functionality, at first we could not use just one of them because of the restriction that a plan can only handle one type of event, while these are different events. This pointed to the insight that they somehow share some basic feature while they differ in some other respect: the situation in which the obligation arises.

The change in data might occur without any of the stakeholders being aware of it. They might not even be aware of the data's existence. This makes the stakeholders more dependent on the proper execution of the obligation to inform. There exists an asymmetry regarding the information. This would be even more pronounced in case only some of the stakeholders are aware the data exist.

New notions

During implementation we were confronted with the challenge of providing simple overviews of what was going on in our experiments. Agents were communicating, exchanging information, updating their knowledge base, and so on. In order to keep track of all these interactions and this data exchange we developed, applied and tuned some notions we believe to be of theoretical and practical use: a) moral Chinese walls and b) deontic isographs with communication tracking.

Ad a) Using the notion of sphere we can define domains in which knowledge should (not) become available. Based on the role of an agent, he can be assigned to a domain, and has or does not have the right to obtain a particular piece of information. The domains are separated by a kind of 'Chinese wall' that should not allow information to filter through. Using this notion we can check whether the set-up of rights, obligations and spheres are functioning as they should. If information is obtained by an agent in another domain (on the other side of the wall) it becomes clear that something is working out in a different way than intended.

Ad b) Continuing the approach from A, and the visual presentation, one sees a pattern develop. Agents that have access to particular information form together a deontic and epistemic isograph, that is a series of an epistemically and deonti-

cally equally endowed agents. Using visual means to track the communication between agents we introduce a new tool for analysis.

Figure 13 illustrates the notions introduced in this section[10]. The different spheres as described in section 4.1 are presented with the functionaries in the role they have. Highlighted areas indicate which functionaries have knowledge of a particular data. These elements form, as it were, a deontic and epistemic isograph.

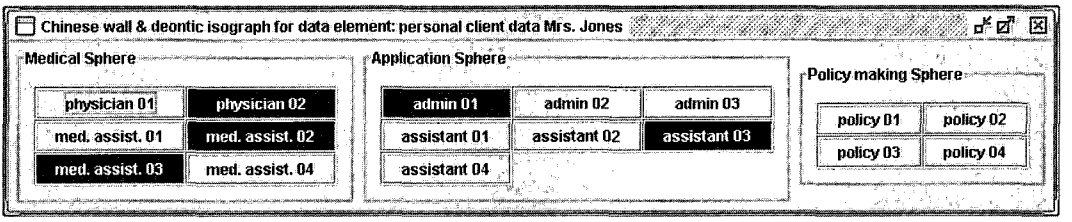


Figure 13: Visual tracking: Chinese walls and deontic isographs

The separation of the spheres provides an analogue of a Chinese wall, i.e., a separation of spheres through which information should not be exchanged. In this case personalized data can be exchanged between the administration and the medical departments of the insurer, but not with the policy making department.

4.6 Conclusion

Our experiments indicate that implementing a DEAL logic is possible. All morally interesting concepts can be implemented in executable code that reflects the theory and operates at a level that is sufficiently detailed for both research and application purposes. The expressiveness is rich enough to capture all relevant moral and informational notions. The challenge is in explicating these notions and putting them into a coherent whole, which involves much work. But rather than holding this against our approach, we think the approach helps in identifying and articulating what we need to express in order to have a complete theory of moral constraints in complex informational relationships.

There are two noticeable potential constraints or limitations of our approach which will require further research[11]. On the one hand, the original BDI model as Bratman presented it seems ill-suited to capture some kinds of behaviour, in particular, learning and adapting. But, of course, embedding learning and

[10] We present here a conceptual prototype that is not yet integrated into our experimental environment.

[11] We would like to thank Philip Brey for some insightful comments and suggestions for further research.

adapting in systems as ours, is crucial in any "ethical" context. It would be interesting to investigate how the specific model provided in the paper could be extended to incorporate adaptive behaviour.
On the other hand, the BDI model embeds "anthropomorphic" assumptions, such as intentions, that might seem inappropriate in systems with artificial agents. We have not touched on this issue in our article. Future work will need to give a proper understanding of these assumptions in an artificial context, or replace them by more suitable notions, such as goal-directedness.

We discussed how our approach might be applied to insurance industry where private, medical data are processed and privacy issues can arise. In addition, we think it suitable for exchange of medical data between providers of medical services, such as general practitioners, apothecaries, and surgeons. There is a lot to say in favour of a persons medical data being online (electronic patient file) accessible for many different providers of medical service, especially in case of emergencies. On the other hand such an availability would pose some serious risks for the privacy. A third potential application can be seen in the financial and business services industry. Some companies act as both account and investment manager for its clients. Information obtained in one capacity might influence the behaviour in the other. E.g. helping a company to issue new shares (and getting paid a big fee for this service) might influence the opinion formed by the accounts ('company is not doing so good', information which would influence the share price and hence the fee).

Although our findings are not final or conclusive yet, and the work is still very much in progress, our understanding of the relevant themes has increased. Some new issues have come up and others are better understood. We do not want to argue that these results could only have been obtained through this approach. We do argue, however, that the approach has contributed to achieving the results and is particularly appropriate. Our approach stands in a new tradition that merges computational facilities with philosophy, a tradition that was started by Thagard (1992), Bynum (1998, 2002), Castelfranchi (1995) and Danielson (1992, 1998), and shows ample opportunity for further extension.

Our approach shares some commonalities with other (non-)moral research. Governatori (2002), for example, uses software agents to implement negotiating strategies for electronic commerce. The basic agent design is similar to ours: sharing finite state machine mechanisms in defining the plans and strategies; the modular components; the use of executable, distributed software; formal logic to express the reasoning schemes. The subject-matter, however, is very different. The subject-material that we studied has also been studied by Broersen (2001) and Dastani (2001), who investigate decision making in a BDI context

extended with obligations. Our emphasis is on the execution and experimentation whereas theirs was more formal and oriented towards decision-theory.
In conclusion, we think there is both a sufficiently rich theoretical basis and sufficient implementational evidence to warrant proceeding along the lines of the present approach.

5 Voting: testing rule- and act-utilitarianism in an experimental setting

Abstract

Computers and computer programming are increasingly used by social scientists and philosophers to support their research activities. The study of utilitarianism can benefit from these new means of research. Utilitarianism is indeed particularly well suited because many assumptions and axioms can and have been cast in logical and mathematical forms. Constructing computer models of a theory introduces the ability to experiment. A methodology, rules and tools and techniques together form a laboratory for philosophical experimentation: SophoLab. The benefits of such an approach include increased rigour in argumentation enforced by the formal nature of computer programs; a powerful data processing device to deal with the inherent complexity that arises from the interaction of many entities; and the ability to test a theory that cannot be tested in real life because it would possibly inflict harm if proven wrong, or because it is impracticable.

To substantiate and illustrate my claim I set up and run several experiments on a part of Harsanyi's discussion of utilitarianism: his argumentation favouring rule utilitarianism over act utilitarianism (Harsanyi, 1982). First, I analyse his theory and deconstruct it into smaller elements that I then translate into a computer setting. While doing so it becomes clear that his theory is underspecified. I fill in the blank spots through parametrized variables. This allows me to switch them on and off, and assign them a whole range of different values that represent different theoretical positions.

It will show that information and its costs are key in the effectiveness of act and rule utilitarianism. Many authors have already drawn attention to the importance of information, and the lack thereof (see for example Hardin 1988). Using my experimental setting the effect of information and its cost can be analysed in new ways. I also identify three further elements that have particular influence on the effectiveness of both strands of utilitarianism: 1) the group size in which individual agents operate, 2) alternative decision-making algorithms that take account of uncertainty, and 3) an inclination (which I broadly interpret as social mores or culture) towards particular types of actions given uncertainty.

The experiments suggest several tentative conclusions and avenues for further research. a) At least with respect to information act and rule utilitarianism are extremes on one continuous scale of assumptions about the cost of information ñ and functionally equivalent in this respect. b) Rule utilitarianism is much more egalitarian than act utilitarianism. c) As the group size grows act utilitarianism performs better in terms of utility generated. d) As institutions grow more wide spread (and I regard institutions as a form of rules in the rule utilitarian sense) and hence the cost of information diminishes act utilitarianism performs better. This is a paradox that act utilitarianism thrives where the effects rule utilitarianism are most noticeable. e) Slight adaptation of Harsanyi's act utilitarian decision rule improves the act utilitarian effectiveness dramatically. f) Inclination towards particular types of actions in uncertainty has a large impact even after information has become available.

The chapter is organized as follows. In section 5.1 I prepare the experiments by analysing Harsanyi's theory of utilitarianism conform the methodology described in chapter 2. In my attempts to translate his theory to the experimental environment, section 5.2, it becomes clear that there are some white spots that have to be filled in before the experiment will be executable. Filling in, and parametrizing these white spots is done in section 5.3. In this section I also introduce the executable experimental environment and match the elements from the theory to the elements in the experimental environment. The results from running the experiments I present in section 5.4. In the final section, section 5.5, I present some tentative conclusions from the experiments.

5.1 Harsanyi's theory of utilitarianism

This chapter follows closely the methodology outlined in chapter 2. This means that the steps described in chapter 2 as part of the methodology will be followed. As the first step in the experimental approach the theory is examined and decomposed. I take Harsanyi's presentation of utilitarianism, and in particular his discussion of act and rule utilitarianism, as my starting point. There are several reasons for this choice. He draws several, very clear conclusions. This clarity helps in defining predictions that can be compared to the outcome of the experiments. His reasoning is concise and allows for a good analysis that is required to decompose his argumentation. He illustrates his argumentation with an example. In experimenting it is critical to have an actual application of the theory to a concrete example (as opposed to abstract theorizing)[1]. Finally, the

[1] By this is not meant a concrete phenomenon as observed in 'reality', but a concrete example. One cannot run an experiment with *an* agent. It has to be agent *xyz*. And that agent cannot just do *something*, it has to do

theme of his argumentation, act versus rule utilitarianism, touches on several debates concerning the extensional equivalence of both forms of utilitarianism, the strategic aspects of co-ordination, co-operation and institutions.

Harsanyi's theory of utilitarianism has its roots in the work of Adam Smith, Emmanuel Kant and utilitarian tradition of Jeremy Bentham and John Stuart Mill. From Smith and Kant he uses the similar concepts of the impartial spectator and the universality principle. Both concepts express in different ways that all should be treated likewise without favouring. Harsanyi reformulates this principle and extends it in his 'equiprobability model for moral value judgements'. From Bentham and Mill he takes the concept of the maximization of social utility (social welfare function) as the criterion of the morally good. He prefers rule utilitarianism over act utilitarianism and formulates his moral criterion as follows.

> "...a correct moral rule is that particular behavioural rule that would maximize social utility if it were followed by everybody in all situations of this particular type." (Harsanyi, 1982: 41)

Finally, rationality plays an important role in his theory. In his view ethics is a part of a general theory of rational behaviour on par with decision theory and game theory. The rationality criterion is expressed as Pareto optimality and the Bayesian rationality postulates.

This all can be formalized as follows. Let S_1, S_2 ,..., S_i ,...., S_z be the strategies of agents 1, 2,.., z, where $\forall i$: $S_i \in \{ s_1, s_2, .., s_t \}$, the set of all possible strategies. The social welfare function, $W()$, is the sum of all individual utilities $U_i(S_i)$, and is maximized over all strategies S_1 to S_z.

$$(19) \quad W_{max} = \max_{S_1 \ldots S_z} W(S) \quad \text{with } W(s) = \Sigma^z_{i=1} U_i(S_i).$$

The utility function must adhere to the rationality requirements such as a complete pre-ordering and continuity. The max is to indicate the each moral agent should chose that action that maximises $W()$.

In my discussion and experimentation I will focus on Harsanyi's preference of rule over act utilitarianism. I will not be concerned with Harsanyi's assumptions on interpersonal utility comparison, his axiomatic justification of utilitarianism nor with the social welfare function. This is primarily a matter of limited scope and space. Hence, I will take many assumptions of his theory for granted

concrete action *a*. One cannot program an agent to lie. One can program agent *xyz* to say "Q", whilst it knows that "not-Q".

whether I agree with them or not. It is my aim to investigate his theory and particular aspects of it, and not argue against it. It is my job as laboratory technician in the SophoLab to set up the experiment with the theory as starting point. The question of rule versus act utilitarianism is the focus point.

Now I turn to the issue of which version of utilitarianism is preferable. Harsanyi (1982: 56) is very clear on this.

> "...the basic question we have to ask is this: Which version of utilitarianism will maximize social utility? Will society be better of under one or the other? This test very clearly give the advantage to rule utilitarianism."

As I have mentioned above, in Harsanyi's view the question of morality is that of maximising social utility. This is the same for both act and rule utilitarianism. Their decision rule, however, differs. For the rule utilitarian agent the decision of other fellow rule utilitarian agents is an endogenous variable, whereas for the act utilitarian agent the decision of all others, be they utilitarian or otherwise motivated, are exogenous.

> "...the two theories impose very different mathematical constraints on this maximization problem. An act utilitarian moral agent assumes that the strategies of all other moral agents (including those of all other utilitarian agents) are given and that his task is merely to choose his own strategy so as to maximise social utility when all other strategies are kept constant. In contrast, a rule utilitarian moral agent will regard not only his own strategy but also the strategies of all other rule utilitarian agents as variables to be determined during the maximisation process so as to maximise social utility. [...] These differences in the decision rules used by the two utilitarian theories, and in particular the ways they define the constraints for the utilitarian maximisation problem, have important practical implications. One implication is that rule utilitarianism is in a much better position to organise co-operation and strategy co-ordination among different people..." 1982, 57)

Where the implication to better organize cooperation arises from is not directly discussed by Harsanyi. But the reasoning will be along the following lines. As rule utilitarian I have a decision rule, assuming that other rule utilitarian agents have the same rule, I can reason about their decision and taking into account the fact that all others will follow the same reasoning and arrive at the same conclusion. But, as we will see later on, there is a lot of assuming behind this argument that might be problematic. The assumptions are: that all rule utilitarian agents involved have the same decision rule, that there are no personal difference

whatsoever, no different motivations, etc. ; that they all know each other, and know each other be rule utilitarian; they all know that they can rely on the other rule utilitarian to execute the same decision rule. To strengthen his point Harsanyi gives an elaborate example.

> "For example, consider the problem of voting when there is an important measure in the ballot but when voting involves some minor inconvenience. Suppose, there are 1,000 voters strongly favouring the measure, but it can be predicted with reasonable certainty that there will also be 800 negative votes. The measure will pass if it obtains a simple majority of all votes cast. How will the utilitarian theories handle this problem?
>
> First, suppose that all 1,000 voters favouring the measure are act utilitarian agents. Then each of them will take the trouble to vote only if he thinks that his own vote will be decisive in securing passage of the measure, that is, if he expects exactly 800 other people favouring the meaner to vote (since in this case his own vote will be needed to provide the 801 votes required for majority). But of course, each voter will know that it is extremely unlikely that his own vote will be decisive in this sense. Therefore, most act utilitarian voters will not bother to vote, and the motion will fail, possibly with disastrous consequences for their society.
>
> In contrast, if the 1,000 voters favouring the measure are rule utilitarian agents, then all of them will vote (if mixed strategies are not allowed). This is so because the rule utilitarian decision rule will allow them a choice only between two admissible strategies: one requiring everybody to vote and the other requiring nobody to vote. As this example shows, by following the rule utilitarian decision rule people can achieve successful spontaneous co-operation in situations where this could not be done by adherence to the act utilitarian decision rule (or at least where this could not be done without explicit agreement on co-ordinated action, and perhaps without an expensive organisation effort)." p 57-58

There are some poignant statements in the example that I will stress and focus on in setting up the experiment.

1) "...by following the rule utilitarian decision rule people can achieve successful spontaneous co-operation..."

2) "...rule utilitarianism is in a much better position to organise co-operation and strategy co-ordination among different people..."

3) "...each of them (act utilitarian agents) will take the trouble to vote only if he thinks that his own vote will be decisive in securing passage of the measure..."

4) "...achieve successful spontaneous co-operation in situations where this could not be done by adherence to the act utilitarian decision rule (or at least where this could not be done without explicit agreement on co-ordinated action, and perhaps without an expensive organisation effort)..."

These statements immediately raise some questions. Questions that I will outline now, and address later when the experiment is set up and run.
Harsanyi mentions admissible strategies (1). This implies the presence of other strategies - the inadmissible strategies. What strategies are these? And what makes the current strategies admissible, and the others not? Without some explanation it is not clear why the rule utilitarian agents reaching some goal deserves the label spontaneous (4)?
Rule utilitarian agents are in a better position to organize cooperation (2) and coordinate their strategy. How does the coordination take place? Coordination and cooperation imply active involvement of agents to agree their actions in order to achieve a common goal. Merely having the same limited set of strategies and decision rule seems to be stretching the notion of coordination and cooperation. If we allow for a somewhat broader notion of coordination and cooperation the question is how that is to take place. Is there someone taking the lead and calling on others? What information flow is required for coordination to be successful? If communication is allowed than both rule and act utilitarian agents should be allowed to communicate. What effect would that have on their behaviour?
In the example there is no co-ordination among the rule utilitarian agents. When they form the majority their coordination skills should enable them to use the fact of their majority to plan additional activities and derive additional utility for those members of the group that are not needed to win the vote.
Assuming that the rule utilitarian agents are able to co-ordinate their activities (2,4) it is not evident from the description why the act utilitarian agents are not able to do the same, or only a great costs (4)? This claim is left unsubstantiated.
A key role in the theories is played by the decision rules. Both seek to maximize utility. The main difference is that in the rule utilitarian decision rule the fellow rule utilitarian agents are endogenous, whereas in the act utilitarian decision rule all other agents' strategies are exogenous. Regarding both decision rules the question is for both endogenous and exogenous variables: what assumptions about the values are made? And how? Do the agents form hypotheses about the

behaviour of others? If so, how do they reason about them? Do they apply some maximum likelihood reasoning? In short, how do they form beliefs about their environment, the exogenous variables in the decision rule?

I hope to have made clear that there are, at first glance, some questions that need answering before we can say that a satisfactory explanation has been given for the claims made. These are not simple questions. And I might not be able to answer them. But then that is not the main aim of this chapter, nor of the methodology. I hope to make clear that these questions pop up when one starts thinking about implementing the theories involved. And, moreover, that in trying to implement them some answers are formed in due course while trying to solve some implementation problems. And last, that through the setting up of generic mechanisms to represent the problems various options can be investigated. E.g. When Harsanyi claims that cooperation and coordination is possible for act utilitarian agents only through an expensive organization the experimenter can set-up an organization mechanism in which the costs are variable. Then running experiments with varying costs the impact can be assessed.

5.2 Preparing the experiment

5.2.1 Step 1: decomposing

The selected theories are rule and act utilitarianism[2] as I discussed them in the preceding section. Their main components are

- Preference/utility function
- A decision rule for individual act utilitarian agents favouring more utility over less

(20) MAX U_i with U_i (S_i | S), where utility U_i is a function of the strategy S_i of agent i, given the set S of strategies of the other agents, i.e. $S=\{S_1,..,S_{i-1}, S_{i+1},\ldots,S_z\}$.

The decision rule for rule utilitarian agents has a similar form but differs in the definition of the endogenous and exogenous variables.

(21) MAX U_i with $U_i = U_i$ (SR | SA), where utility U_i is a function of the set SR of strategies of all utilitarian agents, *given* the set SA of strategies of all act utilitarian agents.

[2] For simplicity's sake I refer to act and rule utilitarianism when I mean Harsanyi's interpretation of them. I am not concerned with the question whether they are a correct representation as my main aim is to show how experimentation can contribute to a debate.

- Situations (outcomes of actions) that can be measured in terms of the social welfare, $W()$, the sum of the utilities of the individual agents, or

  (22) $W() = \Sigma^{z}_{i=1} U_i()$, where $U_i()$ is the utility of agent i.

- Prediction: rule utilitarian agents will outperform act utilitarian agents ñ that is achieve more utility ñ in a given situation: $W_{Rule}() > W_{Act}()$, where $W_{Rule}()$ is the social welfare of all rule utilitarian agents, and $W_{Act}()$ is the social welfare of all act utilitarian agents.

5.2.2 Steps 2 – 4: translating into framework and experimental setting

Step 2 - the example. There are two parties, one act and one rule utilitarian party. They are engaged in a voting that each hopes to win. According to the prediction by Harsanyi the rule utilitarian agents will win the vote whenever they have the majority. And more to the point, the act utilitarian agents will not be able to win, even if they have the majority.

Key in Harsanyi's argument is the fact that each (act) utilitarian has two options: to go voting or do something else (that yields a positive utility). As each of act utilitarian agents has to decide for himself the question he has to answer is: will my voting make any difference? If not, then he will do something else. As each of the act utilitarian agents will think in a similar way none will vote, and the vote is subsequently lost. The rule utilitarian agents on the other hand will all follow one rule, and if that rule is to vote then they will not be in danger of losing it because all will vote.

The act utilitarian agent faces a situation in which there are, logically speaking, four possible outcomes. One, he will not go voting while enough of his fellows will, in which case he derives his share of the benefits from the won vote and the utility of doing *X*. Two, if the vote is lost he will at least have the utility from action *X*. Three, if he votes and the vote is lost he will derive utility from neither. Four, if the vote is won, the winning will provide utility but he has forgone the utility associated with *X*.

To set up an executable experiment the utilities and preference functions have to be exact and quantified. I will use the following pay-off structure for the act utilitarian agents.

do something else while enough others votes	50
vote while enough others votes	40
do something else while not enough others votes	10
vote while not enough others votes	0

Whether this is in line with what Harsanyi has in mind is not entirely clear because he does not elaborate his example further, and leaves his claims and examples unquantified. If not a problem, it is at least an omission. It is important to note that the pay-off structure is parameterized, and can and will be changed in the course of the experimentation to investigate the effect of the amount of utility and the differences in pay-off between the various outcomes.

Step 3 and 4 ñ Translation to the intermediate conceptual framework. The framework is based on the belief desire intention (BDI) model (Bratman 1987, 1991) implemented using the technique of computers and Java programming. It consists of agents that represent the moral agents from the Harsanyi example. As in the example they are involved in a voting. They will have the desire to maximize their utility, they holds beliefs about what other agents will do, and they will form intentions to go voting or do something else. They can reason about their beliefs and intentions in a logical way.

It is here that we encounter several problems. The rule utilitarian agents must decide given the choice of the other agents but excluding their fellow rule utilitarian agents. How do they derive the information about the act utilitarian agents? About their numbers, their intentions, etc.? They can supposedly calculate the benefits of going to vote when they have the majority ñ given that they know their fellow rule utilitarian agents to be rational and having the same information. However, when the chance exists that not all act utilitarian agents might show up when they form the majority, due to their alleged inability to coordinate, it is far from clear how they derive their decision.

It is equally unclear how the act utilitarian agents form their beliefs about the situation they are in. What do they know about their fellows, their decision rules, etc.? Harsanyi assumes that they have certain information.

1) all rule utilitarian agents are the same (that is adhere to the same decision rule),
2) and they know they are the same,
3) and they know about the number of agents involved in the voting,
4) and they know what it takes to win a vote, the qualified majority.

The agents know there are 1800 agents involved, of which 800 belong to one party and 1000 to the other party. They know that a simple majority suffices to win the vote. As how they come to know this and nothing else, Harsanyi provides no explanation. Particularly assumptions 1) and 2) are very strong assumptions to make. It seems very convenient that they have just this information, and no other. And I would maintain that as long as the origins of this information are not accounted for there is little what we would call spontaneity or coordination involved.

Perhaps this is still fine for the rule utilitarian agents, but for the act utilitarian agents not so assumptions are being made. So where do they get their information from? How do they form their beliefs? The act utilitarian agents must decide given the choice of the other agents including their fellow act utilitarian agents. They must have some notion about their fellow agents intended behaviour. How do they derive this information? Will agents assume their fellows are just like them? This will not work because his own intention is dependent on theirs ñ which in turn depends on his, and infinite regress results. Act utilitarian agents cannot be implemented due procedural problems. Whether this is inherent to act utilitarianism is not clear from Harsanyi's discussion. He states they cannot achieve some success without some co-operation.

> "...by following the rule utilitarian decision rule people can achieve successful spontaneous co-operation in situations where this could not be done by adherence to the act utilitarian decision rule (or at least where this could not be done without explicit agreement on co-ordinated action, and perhaps without an expensive organisation effort)." p 57-58

This implies that he does not think act utilitarianism procedurally impossible. Moreover, it seems to me that the procedural problems arise from the fact that no assumptions are made as to the informational situation of the act utilitarian agents.

Harsanyi, apparently, does not exclude some kind of agreement or organization to arrive at some form of co-operation. Now the question is what kind of co-operation would that be? And, why will it be costly? What is costly anyway in the context of this example?

From the discussion in this section it is clear that the argumentation (as presented by Harsanyi) and the example are not implementable as they are[3]. Most crucial is the absence of an explanation how the agents come by some knowledge about the world they live in ñ who their fellows are and what their intentions are.

[3] It might seem childish to focus on the incompleteness of the example and argumentation. It is an article on a very large subject matter. One has to remind, however, that a far-reaching conclusion is based or affirmed by just the theoretical expose and the example as described. And therefore Harsanyi can be criticized on the underspecification. Especially since there are no references to other publications where the deficits might be addressed. A theory that is not implementable is not necessarily wrong. But it is deficient.

5.3 Modelling further

5.3.1 Steps 2 – 4 repeated: extending and adjusting

In the preceding section it became clear that the argumentation of Harsanyi is underspecified. In order to proceed I will try to fill in the gaps. Since I do not wish to take a position in the discussion I will try to fill the gaps in a way that leaves me as much freedom as possible in adjusting the assumptions. Different assumptions about the value of a parameter reflect different positions in the discussion.

So where do both the act and rule utilitarian get their information about other agents from? Remember, that their strategies are conditional on the what the other agents will do, or more accurately on what they believe other agents will do. Harsanyi nowhere describes how this aspect should be addressed. If there is no information flow it is not clear how the decision rule will function. Has each agent to expect other agents to be like himself? Generate a choice at random? We need to do something about belief formation and intentions[4]. It seems reasonable to allow in principle some kind of information flow on all sides. But how will that happen? And what will it cost?

First, since this is about experimentation, adding the possibility of information flow does not mean we will have to use the option. It will allow us to find out what happens in different circumstances.

How then should this information flow take place? One of the options that poses the least demands on institutional organization is having the voters meet in chance groups before the vote, where they can (if they feel like it) question each other about both their type (rule or act utilitarian) and their intention (voting or not). This requires no pre-organized meetings, no knowledge of the whereabouts of fellows, no deliberate grouping (though that would certainly make sense). Moreover, it seems natural, members of a congress, members of parliament meeting each other in the corridors, the lunch room, etc. discussing the upcoming vote. I maintain that this construction is epistemically, ontologically and institutionally less demanding than the assumptions (all rule utilitarian agents have the same decision rule, know each other to be fully rational, know each other to be rule utilitarian) Harsanyi is making .

[4] Beliefs and intentions play an important role in several investigations on rationality and artificial intelligence. It is a general contention that for rationality to be possible an agent must have beliefs, desires and intentions and the ability to reason about them. See Pollock, Bratman, Wooldridge. Rationality is a claim made utilitarian theories. And therefore they need to address these concepts as soon as the functioning of individual utilitarian agents is considered.

Information might or might not be free. To add to possibility of information that is not free I introduce a negative utility on enquiry. Agents are free to decide whether or not to acquire information. The utility (cost of information) will be set at ñ2. This is starting point, and to some extend an arbitrary value, that can be changed during the experiments.

At the start the agents will not have any clue about what to expect, that is, they hold no beliefs about other agent's intentions. They all have an innate desire to maximize utility. They go about gathering information by asking their fellows about their intentions. Now the question is where does this first intention come from. If each agent asks the other agents about their intention, needing this information to form his own intention, we are stuck. I therefore assume some propensity to go voting. We can interpret this as liking, upbringing, chance, etc. This propensity will be parametrized so that its influence can be investigated. The propensity to go voting or not is assigned randomly to the agents.

Using the thus gathered information agents can form their beliefs about the number of voters on both parties. In its most basic form this can be done by extrapolating the findings in the small groups to the whole population. Given the beliefs how will the act utilitarian agents form their final intention that is transformed into action? To start with there are three different decision rules that come to mind.

1) If the agent thinks he will make a difference go voting, otherwise do not go voting.
2) If the agent thinks his party will win anyway or lose anyway do not go voting but use the opportunity to gain some additional utility by doing something else
3) If the agents thinks his party will win keep with the intention because that is an apparently winning intention

All three rules use the same information: expected voters in relation to required number of voters to win:

$$(23) \qquad \alpha - x < EV1 < \alpha + y,$$

where EV is expected number of voters of party one excluding the agent's own vote, α is the majority vote and x, y are tolerance margins. Tolerance margins are used to define a range rather than a point value for which a vote is won or lost. It expresses the uncertainty of the number of actual voters.

Rule 1 (Harsanyi's version)

(24) Vote iff $\alpha - x < EV < \alpha + y$,
otherwise do not vote, where $x = 2$ and $y = 0$

Rule 2 (generalized version of rule 1)

(25) Vote iff $\alpha - x < EV < \alpha + y$,
otherwise do not vote, where $x, y \in \mathbb{R}$ and $\alpha - x, \alpha + y \in [0, 1800]$

Rule 3

(26) Stick with intention to vote or not vote) if $\alpha - x < EV < \alpha + y$,
if $EV >= \alpha + y$ do not vote
if $EV =< \alpha - x$ do vote, where $x, y \in \mathbb{R}$ and $\alpha - x, \alpha + y \in [0, 1800]$

Rule 1 is Harsanyi's version of the act utilitarian decision rule. If, and only if, the agent's vote is decisive, that is he is voter 901, he will go voting. Rule 2 is a generalized version of rule 1. The only difference being that the margins within which the agent will conceive his vote as decisive is extended. He does not have to be voter 901 in order to go voting but might be say voter 904. A justification for this extension is uncertainty. Under circumstances it might be hard to get a correct impression of the exact number of voters. One might get the impression wrong by some margin. This margin then has to be taken into account. Rule 3 is different. It is introduced as an alternative in the experiment to compare against the hypothesis. It says that everyone has an intention to go voting or not. This intention is then matched against the expectation of the number of voters, as in rule one and two. If the expectation is that the vote will probably be won the current set of intentions is perfectly suited and should not be altered. Again, there are some margins for uncertainty. If the expectation is that the vote will be won by a sufficiently large margin the agent will decide to do something else. If the expectation is that the vote will be lost this has to be remedied by going to vote. This is probably the hardest element in the decision rule to justify from a rational point of view. The chance that the vote will be decisive is small indeed, the utility of doing something else is guaranteed and should be preferred. The obvious reaction is, of course, that if the difference in utility is large enough to offset the small chance. Harsanyi allows for this large difference in utility pointing to the disastrous consequences.

> "But of course each voter will know that it is extremely unlikely that his own vote will be decisive in this sense. Therefore, most act utilitarian vot-

> ers will not bother to vote, and the motion will fail, possibly with disastrous consequences for their society." (1982, 58)

If the consequences are disastrous indeed, it is not unrealistic to assume a very large difference between the utility derived from winning the vote and from doing something else. Even more so if the number of agents involved is smaller. The usual parliament does not have 1800 members but something in the order of 75 to 500. In which the chance of a decisive vote is higher.

The point that is made in favour of experimental philosophy, is that matters are often too complex to oversee everything. Hence the need for experiments in which one can try various possibilities and find out how they work. And that is also my main justification for introducing this decision rule alongside Harsanyi's rule. It does not seem beforehand completely unrealistic. Therefore let's give it a try!

5.3.2 The SophoLab software framework[5]

In this section I provide an overview of the elements in the framework that enable the modelling of situations like the voting example. In the step five the general concepts like agent, game, strategy are implemented in an artificial environment where the experiments are run. In this experiment all the below elements are implemented as a computer program. The program is written in the Java programming language and runs on a computer. Through parameter files the settings of the experiment can be adjusted[6].

5.3.3 Step 5: Implementation – elements in the framework

The SophoLab framework[7] contains a set of elements[8] that allows us to construct an experiment as described above. The main elements are:

---

[5] This software framework predates the versions presented in earlier chapters. It is a forbear of the agent software later developed. Therefore the presentation and naming of some elements might appear slightly different.

[6] Besides the elements discussed in this section the program contains additional elements to deal with technical issues. The results of the voting rounds together with the accumulated utility per individual agent are written to an output file.

[7] This framework referred to in this version is a prototype. That is, it is still under development. The following descriptions refer to what is implemented in the prototype unless stated otherwise. So all reference are to working material – functioning software code.

[8] It is important to keep in mind that my approach is instrumental. I model and create elements that take the form of real life elements and perform accordingly but they are emphatically not those elements. In fact they may in their internal functioning differ strongly, as long as it does not create any outward differences. So,

- agent
- group
- game
- referee

Agents

Agents embody the theories. They carry out the strategy that forms the core of the theory by executing actions. From the descriptions above I derive a number of attributes of the agents. They can ask and answer questions after their type. In order to be able to answer questions they must be aware of their own identity. They can recognize their fellows. They can query their environment including the game about their characteristics. This implies some basic knowledge about their environment. They know they are playing a game (our conceptual equivalent of being involved in some human interaction) and can ask questions about the game (our conceptual equivalent of investigating a situation).

Given the information they obtain and process, they form and hold beliefs regarding their environment and their own situation. Reasoning from their beliefs they form intentions that are driven at realising their desires (goals). They can answer questions after their intention and ask others about theirs. They have a memory about these elements. In the current setting there are two agent types: act utilitarian agents and rule utilitarian agents. The decision rules for both types are as described in sections 5.2 and 5.4.

Group

A group is set of agents with at least one member. All agents that play the game form a group. Information exchange and voting are two different interactions that take place within different groups. The groups are independent of each other. They can have different sizes. Typically, the groups for the information exchange are much smaller than the voting group (which consists of all potential voters). The information exchange groups are assembled at random. Agents can be added to groups as a member. Agents decide whether they want to participate in the information exchange and voting and sign up for a group. In the experiment all act utilitarian agents will participate in the information exchange. So will the rule utilitarian agents. There is a difference however. The rule utilitarian agents are just being social. They do not need the information (when they form the majority) because their decision rule tells them what to do independent of the information they might gather. Act utilitarian agents will pay for the information they gather in the information exchange. The rule utilitarian agents

elements may seem and be unrealistic as long as they perform recognizably similar to their real life counterparts.

participate for free (otherwise they would refuse to participate in the information exchange). For simplicity's sake rule utilitarian agents will always participate in the voting game whereas act utilitarian agents might or might not participate depending on their beliefs. Another option is, of course, the rule utilitarian agents never participating in the information exchange. Scope considerations only have kept this option out of this version of the experiment.

This discussion points to an open spot in Harsanyi's presentation. Rule utilitarian agents are not able to exploit the majority they have. The surplus of voters cannot be used to gain additional utility doing something else. Whereas the act utilitarian agents are able to do so. Were the same option open to the rule utilitarian agents they would be prepared to pay for the information as well

Game

A game consists of a set of strategies and the pay-offs associated with the combinations of strategies followed by the players. The number of players per game can both be fixed or variable. In the experiment for both games they are variable. The game entity takes care of the strategies chosen by each agent. It determines the consequences of the combination of strategies as played. How many agents did go voting, who won, etc.? After all agents have made their strategy known it calculates the outcome and assigns utility to each agent in accordance with the strategies played.

Referee

The referee is the entity that enables the running of the experiment in an artificial setting. It instantiates all other entities (games, groups and agents). It represents the environment and keeps track of the runs of the experiment. It is also used as an intermediate for communication among the agents.

5.4 Running the experiments

5.4.1 Steps 6 & 7 – Configuring and running the experiment[9]

Running the experiments consists of executing the computer program with different sets of the parameters. The model embedded in the software has a set

[9] A personal note: I found myself getting more and more involved and curious when running the experiments and changing the parameters than when I was mechanically solving some equations of a mathematical model. I saw trends appearing and trying to give an explanation for the observed behaviour and outcomes. This is, I assume, not unlike a biologist waiting for the results from his experiments.

of input and output variables. The values for these variables define the actual experimental setting.
The input variables are listed in table 4. The output variables are listed in table 5. A run consists of two opportunities to gather information and one voting. For each observation several runs are executed to have a sufficient number of observations that cancel out large deviations that would be due to chance.

Decision rule		Number of runs	
Tolerance margins		Number of agents	
Agent types		Number per type	
Initial inclination		Group size	
Sequence		Pay-off matrices	

Table 4: Input variables

The settings and the outcomes will be summarized in tables as follows

Average score (act)		Max, min score (act)	
Average score (rule)		Max, min score (rule)	

Table 5: Output variables

I will present a summary of the results of many runs with different parameter values. Before doing so I will illustrate the proceedings by taking two runs as example. Table 6 below represents a run in which 80 agents, of which 48 act utilitarian agents and 32 rule utilitarian agents, are involved in a voting. The act utilitarian agents follow decision rule 3 with tolerance margins of 3.5. This means that if they expect between 37,5 (38) and 44,5 (44) of their fellows to go voting they will stick to their original intention. Gathering information costs 1 utility and is done in groups of 40 agents, winning the vote brings 40 utilities per agent, etc. On average the act utilitarian have a score of 243 which is slightly more than the rule utilitarian could have achieved. The maximum average utility for the rule utilitarian agents is 240 (6 times 40 for winning the vote when they are the majority party). The act utilitarian that was best of (won the vote without going to vote) had a total utility of 288 while the worst off scored 228.

Decision rule	3	Number of runs	6
Tolerance margins	3.5, 3.5	Number of agents	80
agent types	Act, rule utilitarian	Number per type	48, 32
Initial inclination	0.8	Group size	40
Sequence	Enquiry Enquiry Voting	Pay-off matrices	Enquiry 1,0 Voting 50, 40, 10, 0
Average score (act)	243	Max, min score (act)	288, 228
Average score (rule)	0	Max, min score (rule)	0, 0

Table 6: An example run

Following decision rule 2 with wider tolerance margins shows results as presented in table 7. The total number of agents is only 20 now. The group size in which information is gathered is smaller (25% of the total group size). On average the utility gathered by the act utilitarian agents is slightly less than what rule utilitarian agents would have achieved.

Decision rule	2	Number of runs	6
Tolerance margins	5, 5	Number of agents	20
agent types	Act, rule utilitarian	Number per type	12, 8
Initial inclination	0.8	Group size	5
Sequence	Enquiry Enquiry Voting	Pay-off matrices	Enquiry 1,0 Voting 50, 40, 10, 0
Average score (act)	235	Max, min score (act)	248, 228
Average score (rule)	0	Max, min score (rule)	0, 0

Table 7: Another example run

Many runs were executed in which some variables were kept constant (e.g. tolerance margins, group size, population size, etc.) while one or two other variables were varied (e.g. inclination) to investigate its success. By running many such test a picture arises that I will now describe.

5.4.2 Decision rule and tolerance

The experiments show that independent of the configuration decision rule 1 is disastrous for the act utilitarian agents. They never win a voting, not even when they form the majority, as predicted by Harsanyi. When the decision rule is relaxed in order to include uncertainty the act utilitarian agents fare better. In some runs they are able to win the voting. Important seems to be the tolerance in the decision rule, that is the extent of uncertainty allowed for. Decision rule 3 is even more successful. From a fairly small tolerance onwards the act utilitarian agents are able to win the vote when they have the majority.

All decision rules allow the act utilitarian agents to exploit the majority surplus. Part of the population does not vote while the vote is still won. In cases where the vote is lost, still some utility is gained by some (rule 2 and 3) or all act utilitarian agents (rule 1).

The tolerance margin can vary from zero to half the size of the population. With a tolerance of zero the decision rule is the one proposed by Harsanyi. With a tolerance of half the population size we have effectively a rule that says 'vote always', this is, of course, the rule utilitarian strategy. As Harsanyi predicted with a tolerance of zero act utilitarian agents are not able to win a vote. This will not surprise. What did surprise was that after an increase to about 3.5, act utilitarian agents are almost always winning the vote when they have the majority. This threshold seems to be independent of the size of the total population, and of the group with which they exchange information. Though for the information exchange to be effective there is a minimal group size.

5.4.3 Cost of information

Another important element is the cost of information. From the previous aspect of tolerance we learned that some tolerance in the decision making helps. This is, of course, only the case if there is some information about what to expect. Thus information exchange is vital. Information is valuable only if it helps increase the chances of a won vote, which again is in part dependent on the tolerance. As the cost of information increases act utilitarian agents still win their votes, but at an increasing cost. When cost is high rule utilitarian agents do markedly better because they have less need for information. This relationship is directly proportional.

5.4.4 Inclination

Inclination to vote or not vote does have a slight impact. The average utility is slight higher with a higher inclination to vote. But even with an inclination of

zero act utilitarian agents win the vote when they are the majority and follow decision rule 3. This makes decision rule 3 very stable.

5.4.5 Step 8 – Translating back to the theory

The decision rule as described by Harsanyi works out badly for the act utilitarian agents. The generalized version (decision rule 2) works already better while the adapted version (decision rule 3) proves even more beneficial. I argued above that the adaptation of the rule does not violate the act utilitarian character, but does take into account uncertainty (which is left out of Harsanyi's account). So with a slight relaxation of the decision rule act utilitarian agents win the vote though the rule utilitarian agents would have done better if that had had the majority (but the point is the act utilitarian agents can win the vote ñ contrary to Harsanyi's prediction). And under certain conditions ñ larger tolerance margins - act utilitarian agents perform better than rule utilitarianism could have done. This follows from their ability to exploit the surplus of votes.

The size of the informal group that exchange information influences the performance significantly. The relationship is not linear. Small (12,5% of total population) and large (50% of total population) groups perform clearly better than medium sized (25% of total population) groups[10].

As the population size grows act utilitarian agents improve their performance. With smaller population they underperform rule utilitarian agents slightly (that is score slightly less than 240 utility) while with a growing population they outperform them slightly.

For rule utilitarian agents the minimum and maximum scores are always the same. For the act utilitarian agents the minimum and maximum score vary markedly. The individual differences are explained by the random influences that are built in through both inclination and grouping. The decision rule appears to be fairly stable to variations in propensity to vote among the act utilitarian agents.

5.4.6 Step 9 – Conclusions regarding the theory

There are stable decision rules that allow act utilitarian agents to function successfully with a minimal requirement of information. The situations in which act utilitarian agents outperform rule utilitarian agents are by no means artificial. The success of act utilitarian agents depends to an important extent on the availability and costs of information, and on the decision rule.

[10] Considering the small set up of the experiment it is mere speculation, but this effect might in an evolutionary context explain why different social groups do, or do not, survive.

Contrary to Harsanyi's claim act utilitarian agents can successfully and spontaneous coordinate their actions. This requires a somewhat different decision rule. This rule, 'do not change a winning intention, change a losing intention', must be further researched to qualify its status. Is it rational for an individual agent to act in this way? From the practical point of view it is, the votes are won. And I think it is rational in the sense that it takes uncertainty into account and allows for unexpected things (fellows falling ill, missing trains, etc. due to which they cannot vote) to happen without immediate disrupting effects.

The cost of information (relative to utility gained from it) is the second important aspect in the debate between rule utilitarianism and act utilitarianism. When information is cheap act utilitarianism performs better than rule utilitarianism. The distinction between both strands of utilitarianism is in the discussion more pronounced than seems justified. As in neither version information and its costs are taken into account the divide is increased. Taking information into the equation actually shows that they are two versions along the same dimension with differing assumptions about the cost of information. They implicitly make different assumptions about information and its availability. In the rule utilitarian case the assumption is that information is scarce or hard to come (that is expensive) relative to the utility gained. Act utilitarianism on the other hand operates from the assumption of information transparency (information is known or can be gained relatively cheaply). Of course, neither of these assumptions holds universally true in real life. It is a mixture. Depending on the situation following a rule or reconsidering it makes sense. This is shown by the experiments described above. If information is cheap the act utilitarian agents perform better than the rule utilitarian agents and vice versa. Both versions are functional extensions in this respect rather than fundamentally different.

A possible venue for further research is an utilitarian theory in which rule and act utilitarianism are combined. As basis rules are followed, but these are checked and deviated from if calculation shows deviation profitable. Since rules are followed generally they do not lose their information value. On the other hand reconsidering allows for innovation.

Pollock (1995) introduces the notion of Q&I modules (Q&I stands for Quick and Inflexible). These are modules that are internalized knowledge rules that can be employed without any explicit reasoning on our part. They have grown out of experience. An example is our ability to catch a ball thrown at us without having to calculate its velocity, the distance between us and the ball, the time the ball will arrive at the place where we are standing, etc.

Rule utilitarianism might very well be interpreted as an ethical Q&I module. There are situations in which we do not have to reason why a particular action is morally wrong, we know it is. But then again there are situations in which we

need ratiocination (in Pollock terms this is the combination of epistemic and practical reasoning, which is relatively slow but explicit) about the moral evaluation of a situation[11].

There is a rule utilitarian paradox. From the previous conclusions it became clear that act utilitarianism thrives in situation of relative information transparency. Information transparency improves with institutionalization. Rules are institutions. So the situations which rule utilitarian agents have helped shaping become favourable for their act utilitarian counterparts.
Act utilitarian agents can better investigate and deal with new opportunities. Since in the survival both absolute and relative score do matter. And though act utilitarian agents perform worse in the voting game then the rule utilitarian agents would have if they would have been the majority, still the act utilitarian agents score better (in absolute terms) because they can exploit the new opportunities.
As a surprise to me came the outcome that rule utilitarianism is egalitarian while the act utilitarianism is much less so. This is due to random influences or individual difference, whatever the interpretation you give. The outcome remains however.

5.5 Conclusion

The experiments show that act utilitarian agents need not fare worse than rule utilitarian agents in certain circumstances. Especially, if one takes into action that they can achieve their results by epistemically less demanding assumptions. They are able to exploit the surplus of votes when they have the majority to gain some additional utility. This compensates for their occasional lose of the vote due to imperfect (wrong) expectations about the number of fellow act utilitarian that will show up. Core at this ability to perform fairly well is a small relaxation of the decision rule as presented by Harsanyi. It consists of allowing some degree of uncertainty into the decision rule. I argued that this is making it more realistic.

The experiments I ran are limited in scope and are open to several objections. I will try to address some of the most obvious objections.

[11] It might be argued that this instrumental approach of utilitarianism misses the point that in fact act and rule utilitarianism hold different opinions about what constitutes right and wrong, the good. I would argue that both have the same opinion (namely the maximum utility) and that it is primarily a matter of how to achieve it. In fact, the discussion between act and rule utilitarianism is not a moral discussion but an instrumental or rationality discussion.

One, one might object that rule utilitarian agents are unduly limited in their options. Their decision rule is much more rigid then the decision rule of the act utilitarian agents. Were they allowed mixed strategies the would outperform act utilitarian agents on all accounts. As experimenter I can only admit that the experiment was limited in scope and that adding this option would certainly provide an interesting set of further experiments. From the theoretical point of view I would point out that as the rules become more numerous or more complex with several conditional clauses the distinction between act and rule utilitarianism becomes more and more blurred.
Two, Harsanyi's argumentation is not a full theory and his article was never meant to be complete. Hence the conclusions I draw and my criticism are beside the points he is trying to make. Though I admit the Harsanyi never claims to be complete nor do I expect he has that intention, his claims are very bold indeed. And if one is making such claims the arguments in favour have to be presented. Any lack thereof due to limited scope is not excuse.
Three, the act utilitarian decision rules in which the tolerance margins are large than {0, 2} are sub-optimal for the agents concerned and hence irrational. In the case of higher tolerance margins the act utilitarian agents should expect to lose or win the vote irrespective of his own vote and therefore do something else. The expected utility would be higher. My first reaction is that as expression of uncertainty it can be rational to use higher tolerance margins. The expected utility is still maximized but now takes into account the possibility that the information the agents possesses might be incorrect. Consequently the uncertainty should be integrated in the function. Another rejoinder would be that the sequence of runs shows the strategy to be better over a series of runs. And when one is concerned with relative scores rather than absolute scores (having more than the others rather than have the highest possible with the risk of having nothing) the strategy is actually doing fine.

Though none of the conclusions and observations I made are conclusive I hope to have shown that setting up experiments is a useful way to gain new and deeper insight in existing argumentation. Moreover, I think several of the observations are worth pursuing.

6 Conclusion

Can we construct autonomously acting agents that we can trust to make morally justifiable decisions? This is one of the driving questions for the research of this thesis. If positively answered this would be very welcome in at least three respects. Firstly, we witness a technology driven development in which we rely more and more on artificial constructs to act on our behalf. This is surrounded by uncertainty about the trust worthiness of the agents, and the risk of fraud and misuse. If we can demonstrate that a particular artificial agent can provide a justification of its actions in which moral considerations play a role, and act accordingly, it would greatly enhance our trust in these agents. It is not the aim of this research to construct these agents. The aim is to investigate if, and how, moral reasoning can be implemented in artificial agents.
Secondly, these very same agents would provide an ideal testing ground for moral theories. Moral theorising is often restricted by the limited availability of empirical data. This restriction could be partly elevated by providing experimental results from tests of theories.
Thirdly, in setting up experiments philosophers will be forced the explicate their theories more rigorously than usual. As the means for experimenting will probably include tools such as computer programs, the instructions will have to be very precise and complete. This rigour, enforced by the experiments, will be a help in theorizing as it will help identify blanks spots in the theory. Another benefit is that the agents will enable philosophers to construct experiments to investigate complex situations. Complexity that can no longer be addressed from the proverbial armchair, but which requires active involvement in constructing models that can be run on computers.
With these broad claims in mind this chapter will review the research presented in this thesis. The research goals, hypotheses, deliverables and questions, as presented in chapter 1, will be reviewed. As the goals, on the one hand, and the steps that lead to these goals (the hypotheses, questions and deliverables), on the other hand, are closely related, I will start with the latter and derive from these the evaluation of the overarching goals.

6.1 Hypotheses, deliverables and questions

The validation of the hypotheses is given by the success, or lack thereof, of the deliverables, and by the answers given to the research questions.

The hypotheses underlying the current research are the following:

4) there is a common, underlying methodology for experiments conducted in the field of computational philosophy;
5) experimentation more specifically with *ethical* theories is possible and fruitful, i.e. increase understanding of a theory, enhances the insights into the implications of a theory and provides support for a theory;
6) DEAL / modal logic and software agents are well suited for the purpose of experimentation and application, and allow the capturing and implementation of thin, central moral notions such as obligation, right and permission.

The hypotheses can be detailed by the following sub-questions:

1) is there a common underlying approach for the various philosophical research conducted with the help of computers?
2) what does a methodological framework for the use of computers in philosophical research look like?
3) what are the methodological guidelines for researchers involved in experiments in the field of computational philosophy?
4) is the combined modelling capability of the BDI-model and the DEAL framework sufficiently rich to capture central moral notions?
5) can the logical models be implemented using the JACK agent language to provide an environment in which the model can be executed?
6) is it possible to capture, express and implement deontic constraints on informational relationships?
7) can experimentation shed light on long-standing debates in ethics: i.c. the utilitarian research community about the comparative strengths and weaknesses of act and rule utilitarianism?

From these hypotheses the following research goals and deliverables are derived:

1) the definition of the methodological framework underlying the experiments conducted in computational philosophy;
2) the definition of a modelling framework for experiments in moral philosophy;
3) the development of a software tool set that allows the implementation of the moral models in executable code;
4) to illustrate the plausibility and viability of deliverables 1 through 3 via the conduction of experiments.

6.1.1 Methodology of experimental philosophy

Looking closely at the work of Danielson and Thagard, and looking more generally at the research collected by Moor and Bynum, it is clear that a similar approach is followed in experimenting in philosophy. Though the work is very different in nature and subject-matter, this need not surprise. The fundamental aspect of experimenting is the 'translation' of research questions and theories from one domain to another. Questions on morality and philosophy of science need to be rephrased so that they can be made executable on computers. It is this fundamental aspect that directs the philosophers cited towards a similar approach. The next key aspect of translation is that the notions from the theories do not exist in the computerized experimenting environment. There exists not predefined notion of 'theoretical coherency' on the computer in the science philosophical sense as meant by Thagard. Hence, Thagard and other experimental philosophers, have to reconstruct these entities in the new environment using the more primitive entities that are available. The work involved in this suggest its might be useful to develop a standard set of instruments, re-usable components from which philosophers can construct their experiments (see section 6.3).

The similarity found is used to define the framework for philosophical experimentation. This methodology consists of nine steps to deconstruct the theory and the test cases, and then reconstruct them in the experimental setting. After experiments have been run the results undergo the reverse route in order to draw conclusions from the experiments. Though the examples investigated use the computer for experimenting there is no reason a priori to exclude other tools and means of experimentation. Thus, by defining the methodological framework deliverable one, "define the methodological framework underlying the experiments conducted in computational philosophy", is provided. That also answers research questions one "is there a common underlying approach for the various philosophical researches conducted with the help of computers", and two, "what does a methodological framework for the use of computer in philosophical research look like", establishing that there is both such a framework and indicating what it looks like. And so, the first hypothesis "there is an common, underlying methodology for the experiments conducted in the field of computational philosophy" is validated.

Using the framework for three subsequent experiments I demonstrated the plausibility of the methodological framework presented. In three very different experiments, on privacy, utilitarianism and moral commands, each time the steps as articulated in the framework were applicable. This constitutes, together with the detailed analysis of two paradigmatic cases, deliverable four: to illustrate the plausibility and viability of the methodological framework.

In order to be acceptable evidence in an argumentation the experiments need to be freely accessible to other researchers such that they can be validated and repeated. Since it cannot be excluded that a particular framework and/or technique is biased towards a theory it is desirable to repeat experiments in different settings.
In short, experiments must be verifiable and repeatable. In order to achieve this transparency regarding assumptions and configurations are required. Also openness, i.e., accessibility before, during and after the experiments would be necessary. This is, contrary to the framework, a prescriptive aspect of philosophical experimentation. Most research, including the research conducted in this thesis, falls short in this respect. However this may be, it answers question three, "what are the methodological prescriptions for any researcher involved in experiments in the field of computational philosophy". As computational philosophy is still a very new research domain it is expected that much more new insights will be gained in the coming years. These prescriptions can only be a first step towards developing a full-fledged methodology with prescriptions.

6.1.2 Modelling and constructing experiments on moral philosophy

In order to be able to experiment one needs both the modelling tools and the implementation to actually run the experiments. The modelling needs to be close to the application domain, in this case moral philosophy. This implies that it must be able to accomodate notions of agency and goal-directedness. Also, a basic ability to assign values is required. Through Bratman's BDI-model both the directedness, epistemic aspect and the action orientation is covered. To supplement this modelling component the DEAL framework was added. The deontic logic is crucial fo the moral philosophical application. And the action logic extends the intentional aspect of the BDI-model. The epistemic aspect in DEAL and the belief component in the BDI-model overlap. By combining the DEAL framework and the BDI-model two well established components are chosen that cover all necessary grounds for the purposes of this thesis. Hereby deliverable two, "define a modelling framework for experiments in moral philosophy", is given.

This combination is used in chapter 3 and 4 to model morally laden applications. In chapter 3, the nature of moral commands is investigated. Moral prescriptions are a core element in moral philosophy. Using the modelling toolset described all relevant aspects can be covered. In chapter 4, the deontic constraints on informational relationships is addressed. This is an intricate issue where particular agents are related in information exchange that is potentially sensitive, and relates to issues of privacy. It proved possible to capture these relationships and their constraints using the DEAL framework and the BDI-

model. Which answers questions four, "is the combined modelling capability of the BDI-model and the DEAL framework sufficiently rich to capture all moral relevant moral notions", affirmatively.

Being able to model morally relevant aspects is necessary but not sufficient for experimentation. The models obtained must be translateable to an executable environment, that is a situation in which the experiments can actually be run.
As technique for execution programmed computers were chosen. The choice for the software was particularly important. The closer the match to the modelling concepts, the easier it is to implement the models and run the experiments. The choice fell on JACK, an industry strength and state of the art development environment for agent software. It is based on the BDI-model and thus provided a good match for one half of the modelling toolset. Its programming language, Java plus an additional set of action statements, quite naturally provided support for the action logic component, and the other modal logic component. The only element missing was native support for deontic logic. Using the available programming elements this support had to be constructed tailor-made.
That the afore mentioned experiments are not only modelled but also executed confirms deliverable three "develop a software tool set that allows the implementation of the moral models in executable code". It also answers research question five "can the logical models be implemented using the JACK agent language to provide an environment in which the model can be executed".

6.1.3 Running experiments

In the debate on effectiveness of rule-utilitarianism versus act-utilitarianism strong claims are made by Harsanyi. Chapter 5 demonstrates how the attempt to implement a theory brings to light the white spots in the theory. Through parameterizing a number of theoretical positions can be investigated, for example, those regarding the cost of information. In this case the experimentation did not only help to identify underspecification, but it also helped to refine positions. In particular, it showed that rule-utilitarianism and act-utilitarianism can be better construed as lying on a continuum with regard to the cost of information rather than being diametrically opposed. The debate suffered from the underspecification, and appeared to be more pronounced than justified based on the theoretical arguments. This analysis confirms and demonstrates how "can experimentation shed light on long-standing debate in the utilitarian research community about the comparative strengths and weaknesses of act and rule utilitarianism" (question seven). It also constitutes part of deliverable four ("illustrate the plausibility and viability of 1 through 3 via the conduction of experiments").

Likewise, the experiments already mentioned in section 6.1.2, on deontic constraints and moral commands, constitute the other parts of the of deliverable four. The experimentation on the deontic constraints on informational relationships in particular showed the added value of computer aided research in situations where complexity is enormous. A complexity that arises from the large number of agents involved. In the description of the experiment it became obvious that one does not need very complex relationships, or very large numbers of agents, in order to get complexity that is beyond the mere theorizing capacities of the average philosopher, and thus, where some of the added value of SophoLab lies. The experiment positively answered research question six, "is it possible to capture, express and implement deontic constraints on informational relationships".
All the experiments together show that is possible and useful to experiment. New insights come to light and theoretical weaknesses become visible. This is not to claim that these results can be achieved only through experimentation. But it showed that experimentation is a powerful and expedient way to gain the results. To conclude with hypotheses two: experimentation with ethical theories is feasible and fruitful.

6.2 This research and its relevance: some news items

An article in January 2006 by Antone Gonsalves on TechWeb Technology News (www.techweb.com), stated

> "Three of four major search engines subpoenaed by the Bush administration have acknowledged that they handed over search data in the government's efforts to revive an anti-porn law that was rejected by the U.S. Supreme Court.
>
> Microsoft Corp., which owns MSN, Yahoo Inc. and America Online Inc. said they sent data to the government, but insisted no personal information on users was given to government attorneys. The exception among major search engines was Google Inc., which said it would "vigorously" fight the government's requests."

Half a year, august 2006, latter an even bigger event made the news as New York Times (www.nytimes.com) reporters Michael Barbaro and Tom Zeller Jr reported the identification of several people from a pool of supposedly anonymized search queries. These queries became public unintendedly.

> "Buried in a list of 20 million Web search queries collected by AOL and recently released on the Internet is user No. 4417749. The number was assigned by the company to protect the searcher's anonymity, but it was not much of a shield.
>
> No. 4417749 conducted hundreds of searches over a three-month period on topics ranging from "numb fingers" to "60 single men" to "dog that urinates on everything."
>
> And search by search, click by click, the identity of AOL user No. 4417749 became easier to discern. [...]." It did not take much investigating to follow that data trail to Thelma Arnold, a 62-year-old widow who lives in Lilburn, GA., frequently researches her friends' medical ailments and loves her three dogs. "Those are my searches," she said, after a reporter read part of the list to her.
>
> AOL removed the search data from its site over the weekend and apologized for its release, saying it was an unauthorized move by a team that had hoped it would benefit academic researchers.
>
> But the detailed records of searches conducted by Ms. Arnold and 657,000 other Americans, copies of which continue to circulate online, underscore how much people unintentionally reveal about themselves when they use search engines ó and how risky it can be for companies like AOL, Google and Yahoo to compile such data.
>
> Those risks have long pitted privacy advocates against online marketers and other Internet companies seeking to profit from the Internet's unique ability to track the comings and goings of users, allowing for more focused and therefore more lucrative advertising.
>
> But the unintended consequences of all that data being compiled, stored and cross-linked are what Marc Rotenberg, the executive director of the Electronic Privacy Information Center, a privacy rights group in Washington, called "a ticking privacy time bomb."

Whether this is something trivial or not, and a real threat or not, is something for others to determine. It does show that privacy and the use of technologies such as search bots, agents, search engines has potential moral repercussions. The modelling of deontic constraints on informational relationships is directly relevant to the events cited above. More generally speaking, the tools and techniques used in this research will find practical application, and help us reason about the above and other issues.

6.3 Future research

Though successfully achieving the research goals set this research is only a first step. At various stages potential research has been descoped and potential extensions have not been added. I will note a few important topics for future research.
Temporal logic. The modelling toolset provided here does not contain a temporal component. Though temporal logic is available as a standard branch of modal logic it has not been included yet in the DEAL framework. Extending DEAL into DETAIL, as Lokhorst suggested, would address this deficiency.
Teams. The modelling toolset, and the software, do provide support for teams of agents. This research did not address the theme of group or team action. In a single action focus one can very well research the relationship between agents, and the behaviour that results from the interaction. From a theoretical point of view, and in practice, to include joint action of agents in a group, is a relevant aspect. For future research it would be good to investigate this team aspect. Recent research in the field of modal logic shows growing support for team notions.
Embodiment. The agents created in the experiments in this research were not embodied. Falling outside the scope of this research I cannot indicate what impact embodiment would have. However, it seems not unreasonable to assume it could have a large impact. Amongst others, it would effect the epistemic abilities of the agents. New experiments would gain from extension to embodiment, though it might be quite a big step to take.
Epistemology. The agents in the experiments could only function in a limited application domain. The information they receive was strong typed. This means that they know upfront what sort of information it is they receive (not what information). Extending moral epistemology is another fruitful but challenging domain to extend this research to. The theories on moral epistemology might not be quite ready to provide directly implementable propositions and concepts. And thus, stand to gain a lot by experimenting.
Learning. Closely related to epistemic capabilities is the ability to learn. In artificial intelligence research, and agent research learning has received a lot of attention. Adding learning capabilities would be relevant from a moral philosophy point of view. It would strengthen our ability to investigate the evolvement of norms over time.
Learning, epistemology, embodiment and team operation all are closely interwoven aspects. To extend the experimentional apparatus of SophoLab in these respects requires a substantial research effort. An effort that would be challenging and very rewarding.

Toolkit. In setting up experiments philosophers have to construct their our building blocks that artificial agents are made up of. This work is laborious and add little value from the experimental point of view. Researchers waste their time by this duplicate effort. At least where it concerns the construction of basic components like, for example, message exchange between agents, simple knowledge storage capabilities. Because each philosopher constructs his own framework the chances are the at the basic level ontologies divert, and the results from the experiments cannot be compared. It would be useful to define a basic set of ontologies that can be shared broadly across domains of philosophy, like philosophy of science or moral philosophy. Standard implementations of such a set would reduce the effort on part of the experimenting philosopher, and at the same time enhance the comparability.

Literature

Adam, A., Delegating and distributing morality: Can we inscribe privacy protection in a machine?, in *Ethics and Information Technology*, 2005

Adami, C., *Introduction to artificial life*, Springer, New York, 1998

Agent Oriented Software Pty. Ltd, JACK, www.agent-software.com.au

Allen, C., Smit, I., Wallach, W., Artificial Morality: Top-down, Bottom-up, and Hybrid Approaches, in *Proceedings of CEPE2005*, 2005.

Alonso, E., d'Inverno, M., Kudenko, D., Luck, M., Noble, J., Learning in Multi-Agent Systems, *Knowledge Engineering Review* 16(3), 277-284, 2001

Artikis, A., Kamara, L., Guerin, F., Pitt, J., Animation of Open Agent Societies, *Proceedings of the Information Agents in E-Commerce symposium*, AISB convention, pp. 99-109, 2001

Artikis, A., Pitt, J., Sergot, M.J., Animated Specifications of Computational Societies, in *Proceedings of Autonomous Agents and Multi-Agent Systems* (AAMAS), Bologna, pages 1053--1062, 2002

Bahrami, A., *Object oriented systems development*, McGraw-Hill, 1999

Bedau, M.A., Philosophical content and method of artifical life in *The Digital Phoenix*, Bynum et. al. (eds), 1998

Berkeley, I., *Connectionism Reconsidered: Minds, Machines and Models*, Cogprints.org, 1998

Berg, J. van den, Computational intelligence techniques and their underlying dilemmas. In G.O.S. Ekhaguere and C.K. Ayo, editors, *New Trends in the Mathematical & Computer Sciences with Applications to Real World Problems*, Publications of the ICMCS. Lagos, Nigeria, 2007, in press.

Boden, M., *The philosophy of artificial intelligence*, OUP, Oxford, 1990

Boella, G., Torre, van der L., Fulfilling or Violating Obligations in *Normative Multiagent Systems*. IAT 2004: 483-486, 2004

Booch, G., *Object Solutions*, Addison-Wesley, 1996

Bratman, M.E., *Intention, Plans and Practical Reasoning*, Harvard University Press, Cambridge, 1987

Bratman, M.E., Israel, D. J., Pollack, M. E., *Plans and Resource-Bounded Practical Reasoning*, in J. Pollock and R. Cummins, eds., Philosophy and AI: Essays at the Interface, pp. 7-22, MIT Press, Cambridge, 1991

Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J.,Perini, A., *TROPOS: An Agent-Oriented Software Development Methodology*, Technical Report DIT-02-015, Informatica e Telecomunicazioni, University of Trento, 2002

Broersen, J., Dastani, M., Z. Huang, J. Hulstijn, Torre, van der L., The BOID architecture, in the *Proceedings of the fifth international conference on Autonomous Agents* (Agents2001), Montreal, 2001a

Britz, K., Heidema, J., Labschagne, W.A., A modal perspective on defeasible reasoning, paper submitted to *Advances in Modal Logic*, 2006

Broersen, J., Dastani, M., Torre, van der L., Resolving Conflicts between Beliefs, Obligations, Intentions, and Desires. *ECSQARU* 2001: 568-579, 2001b

Bynum, T.W., Moor, J.H. (editors), *The Digital Phoenix*, Blackwell Publishing, Oxford, 1998 (revised 2000)

Bynum, T.W., Moor, J.H. (editors), *Cyberphilosophy*, Blackwell Publishing, Oxford, 2002

Caldwell, B., *Economic Methodology in the twentieth century*, Unwin Hyman, London, 1982

Castelfranchi, C., Conte, R. Understanding the functions of norms in social groups through simulation, in *Artificial Societies*, Gilbert, N., Conte, R., (eds), UCL Press, 1995

Castelfranchi, C., Modelling social action for AI agents, in *Artificial Intelligence*, 103:157--182, 1998

Castelfranchi, C., Artificial Liers: why computers will (necessarily) deceive us and each other, in *Ethics and Information Technology*, pp. 1-7, 2000

Danielson, P., *Artificial Morality*, Routledge, London, 1992

Danielson, P. (editor), *Modeling rationality, morality and evolution*, Oxford University Press, New York, 1998

Dastani, M., Hulstijn, J., van der Torre, L., The BOID architecture: conflicts between beliefs, obligations, intentions and desires, in *Proceedings International Conference on Autonomous Agents*, 2001

Dastani, M., & Torre, L. van der, A Classification of Cognitive Agents. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society (Cogsci'02)* (pp. 256-261). Washington DC, USA, 2002

Dastani, M., Hulstijn, J., van der Torre, L., BDI and QDT: a comparison based on classical decision theory, in *Proceedings of GTDT2001*, Stanford, 2001

Dastani, M., Hulstijn, J., Dignum, M.V., Meyer, J.C., Issues in Multiagent System Development, in *Proceedings AMAAS*, 2004

Dignum, F., Kinny, F., Sonenberg, L., From desires, obligations and norms to goals, in *Cognitive Science Quarterly, Vol.2*, No.3-4, pp. 407-430, 2002

Dignum, M.V., V·zquez-Salceda, J., & Dignum, F.P.M., A Model of Almost Everything: Norms Structure and Ontologies in Agent Organizations. In N. Jennings, C. Sierra, L. Sonenberg, & M. Tambe (Eds.), *Proceedings AAMAS'04* . New York: ACM, 2004a

Dignum, M.V., V·zquez-Salceda, J., & Dignum, F.P.M., OMNI: Introducing social structure, norms and ontologies into agent organizations. in P. Bordini & et al. (Eds.), *PROMAS 2004* (pp. 183-200). Heidelberg: Springer, 2004b

Floridi, L., What is the philosophy of information?, in *Cyberphilosophy*, Bynum et. al. (eds), 1998

Floridi, L. (ed.), *Philosophy of computing and information*, Blackwell, Malden, 2004a

Floridi, L., Sanders, J.W., On the morality of artificial agents, in *Mind and machines*, 2004b

Floridi, L., Consciousness, Agents and the Knowledge Game, in *Minds and Machines*, Vol. 15, No. 3-4., pp. 415-444, 2005

Frank, R., *What price the moral high ground?*, Princeton University Press, 2005

Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.), *Handbook of logic in artificial intelligence and logic programming, Volume 5 Logic programming*, Clarendon press, Oxford, 1998

Georgeff, M.P., Pell, B., Pollack, M.E., Tambe, M., Wooldridge, M., The Belief-Desire-Intention Model of Agency, *ATAL 1998*: 1-10, 1998

Governatori, G., A Formal Approach to Negotiating Agents Development, in *Electronic Commerce Research and Applications*, 1 no. 2, 2002

Governatori, G., Rotolo, A., Defeasible Logic: Agency, Intention and Obligation, *Deon 2004*, 2004

Guerra-Hern·ndez, A., El Fallah-Seghrouchni, A.., Soldano, H., Learning in BDI Multi-agent Systems, *CLIMA IV* 2004: 218-233, 2004

Hacking, I., *Representing and intervening*, Cambridge University Press, Cambridge, 1983

Halpern, J.Y., Moses, Y., A guide to completeness and complexity for modal logic of knowledge and belief, in *Artificial Intelligence* 54:319-379, 1992

Halpern, J.Y., On the adequacy of modal logic, in *Electronic transactions on artificial intelligence*, 2000

Hardin, R., *Morality within the limits of reason*, The Universoity of Chicago Press, Chicago, 1988

Harsanyi, J.C., Morality and the theory of rational behaviour, in *Utilitarianism and beyond*, Sen et al. (eds), 1982

Hoek, van der W., Wooldridge, M., Towards a Logic of Rational Agency, in Logic *Journal of the IGPL* 11(2): 135-159 , 2003

Hoek, van der W., Roberts, M., Wooldridge, M., *Social laws in alternating time: Effectiveness, feasibility, and synthesis*. Technical Report ULCS-04-017, The University of Liverpool, 2005

Hoen, P.J., Bohte, S.M., Gerding, E.H., La Poutre, H., Competitive market-based allocation of consumer attention space,. in M. Wellman, editor, *Proceedings*

of the 3rd ACM Conference on Electronic Commerce (EC-01), 202-206 The ACM Press, 2001

Hoven, J. van den, Lokhorst, G-J., Deontic Logic and Computer Suppported Computer Ethics in *Cyberphilosophy*, Bynum et. al. (eds), 2002

Howden, N., R nnquist, R., Hodgson, A. and Lucas, A., 'JACK Intelligent Agents -- Summary of an Agent Infrastructure', in Proceedings of the 5th International Conference on Autonomous Agents, Montreal, 2001

Hughes, G.E., Cresswell, M.J., *A new introduction to modal logic* Routledge, 1996

Humphreys, P., *Extending Ourselves: Computational Science, Empiricism, and Scientific Method*, Oxford University Press, 2004

Jansen, M.C.W., *Micro and Macro in Economics. An Inquiry into Their Relation*, 1990

Jennings, N. R. , On Agent-Based Software Engineering, in *Artificial Intelligence* 117, pp. 277-296, 2000

Kamara, L., Artikis, A., Neville, B., Pitt, J., Simulating Computational Societies, *ESAW III: Third International Workshop*, Springer-Verlag, pp. 56-67, 2003

Korienek, G., Uzgalis, W., Adaptable Robots in *Cyberphilosophy*, Bynum et. al. (eds), 2002

Lakemeyer, G., Nebel, B. (eds.), *Exploring artificial intelligence in the millenium*, Morgan Kaufmann, 2003

Lawvere, F.H., Schanuel, S.H., *Conceptual mathematics*, Cambridge University Press, Cambridge, 1997

Lee, R.C., Tepfenhart, W.M., *UML and C++*, Prentice Hall, 1997

Lokhorst, G-J., Reasoning about actions and obligations in first-order logic, *Studia Logica* 57, 221-237, 1996

Lomuscio, A., Sergot, M., Deontic Interpreted Systems, *Studia Logica* 75(1): 63-92, 2003

Magnani, L., Computational Philosophy Lab, http://www.unipv.it/webphilos_lab/cpl/, (1994) 2006

Marr, D.C., Artificial intelligence: a personal view, in *The philosophy of artificial intelligence* Boden (ed.), OUP, Oxford, 1990

McCarthy, J., Modality, Si! Modal logic, no!, in *Studia Logica*, 59:29--32, 1997

Miller, K., and Larson, D., Angels and artifacts: Moral agents in the age of computers and networks, in *Journal of Information, Communication & Ethics* in Society, Vol. 3, No. 3, 151-157, 2005

Moor, J.H., The nature, importance, and difficulty of machine ethics, *IEEE* (forthcoming), 2006

Moore, G.E., *Principia Ethica*, Cambridge University Press, New York, 1903

Neville, B., Jeremy Pitt, J., A Computational Framework for Social Agents, in *Agent Mediated E-commerce. ESAW 2003*: 376-391, 2003

Perini, A., Bresciani, P., Giorgini, P., Giunchiglia, F., Mylopoulos, J., Towards an Agent Oriented Approach to Software Engineering, *WOA 2001*: 74-79, 2001

Pollock, J.L., *Cognitive Carpentry*, MIT Press, Cambridge, 1995

Quine, W.V.O. , *Theories and things*, Belknap Press, 1981

Rao, A. S., Georgeff, M. P., Modeling agents within a BDI architecture, in *Proc. of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR '91)*, R. Fikes and E. Sandewall, eds. pp. 473--484. Morgan Kaufmann, Cambridge, 1991

Rao, A. S., Georgeff, M. P., An abstract architecture for rational agents, in *Proceedings of the KR92*, 1992

Russell, S., Norvig, P. (2003) *Artificial Intelligence*, 2nd edition, Prentice Hall.

Sergot, M., Richards, F., On the Representation of Action and Agency in the Theory of Normative Positions, in *Fundamenta Informaticae*, 48 (2-3): 273-293, 2001

Sen, A., Williams, B., *Utilitarianism and beyond*, Cambridge University Press, Cambridge, 1982

Sloman, A., *The computer revolution in philosophy : philosophy, science and models of mind*, Harvester Press 1978

Stahl, B., Responsible computers? A case for ascribing quasi-responsibility to computers independent of personhood or agency, in *Ethics and Information Technology*, 2006

Stegm ller, W., *Haupstr mungen der Gegenwarts-Philosophie* Band II, Alfred Kr ner Verlag, Stuttgart 1987

Sunstein, C. R., *Moral Heuristics*, U Chicago Law & Economics, Olin Working Paper No. 180, 2003

Thagard, P., *Computational philosophy of science*, MIT Press, Cambridge, 1988

Thagard, P., *Conceptual revolutions*, Princeton University Press, Princeton, 1992

Thagard, P., Computation and the philosophy of science, in *The Digital Phoenix*, Bynum et. al. (eds), 1998

Tuomela, M., Hofmann, S., *Simulating rational social normative trust, predictive trust and predictive reliance between agents*, forthcoming

Walzer, M., *Spheres of Justice*, Basic Books, New York, 1983

Wiegel, V. Hoven, J. van den, Lokhorst, G-J., Privacy, Deontic epistemic action logic and software agents, in *Ethics of new information technology*, 2006

Wiegel, V., SophoLab (forthcoming), in *Ethics and Information Technology*, 2007

Williams, B., *Ethics and the limits of philosophy*, Harvard University Press, 1985

Wooldridge, M., *Reasoning about Rational Agents*, MIT Press, Cambridge, 2000
Wooldridge, M., *MulitAgents Systems*, John Wiley & Sons, Chichester, 2002

Epilogue

Drivers for SophoLab

There are at least three perspectives on the combination of experimentation with artificial agents and ethics that make it a very interesting combination. One, attempting to create artificial agents with moral reasoning capabilities challenges our understanding of morality and moral reasoning to its utmost. It requires attention to the general structure and architecture, as well as to the little details. As the computer is a relentless component that will not function until completely instructed it pushes us moral philosophers to explicate everything we know. It also challenges our formal apparatus to express our knowledge such that is formal enough for software engineers to be understood. Two, complexity of the situations to which ethical theories must be applied is growing. Modern society with large multi-national organizations and extensive information infrastructures provides a backdrop for moral theories that is hard to encompass through mere theorising. Computerized support for theorising is needed to be able to fully grasp and address the inherent complexity. Three, with the advancement of technological development artefacts play increasingly important roles in our lives. They do not only contain information about us, they start to act on our behalves. With the increasing autonomy comes an increased need to ensure that their behaviour is in line with what we expect from them. We want to ensure that no abuse is made when opportunity offers, no harm is done to others, etc. Besides this negative, constraining aspect there is also a positive aspect. If we want to develop machines that can act to help us, for example, in circumstances where we cannot come because it is too dangerous, they will need all the reasoning power we can give them including moral reasoning.

When attempting to construct artificial moral agents we need to define, at least roughly, what it is that we are aiming at. Moor proposes a classification of four types of ethical agents: ethical impact agents, implicit ethical agents, explicit ethical agents and full ethical agents (Moor 2006). Ranging from agents that have an impact by their very existence to full blown, human-like reasoning agents that have consciousness, intentionality and an ability to provide plausible justifications for their actions.
Given our current understanding of moral reasoning, artificial intelligence, epistemological devices, etc. the best we can try to construct are explicit ethical

agents that can make some ethical judgments that are not hard-wired into their make-up and have some ability to provide an account of how they arrived at their judgment.

In terms of sophistication we are now looking to implement second generation moral agents. The first generation artificial moral agents were ground breaking in that they for the first time implemented moral decision making (a paradigmatic example in case Danielson, 1992) . They are limited in their orientation (moral issues only), their internal make-up, and interoperability and interconnected-ness. The current generation will have support for forms of interconnectedness (team work for example), be multi-purpose (mixture of moral and non-moral goals) and still be very limited in embodiment (mainly electronic) and epistemology (only strong typed). A next generation will probably bring physically embodied agents with strong moral epistemological capabilities.

Approach

In implementing and experimenting with artificial constructs we follow a methodology that focuses on the translation from one environment - theory and application - to another - the artificial setting. This translation happens in several steps. Key is that the theory and application examples are modelled in a (semi)formal language that is close to the implementation environment. The requirements that follow from this methodology are amongst others:

- neutrality - support for various (meta-)ethical theories
- executability
- comprehensibility
- formality
- configurability
- scalability
- extendibility

(see also Dumas et. al., 2002, who come up with a similar set of requirements for artificial agents geared for economic transactions, and with whom I share kindred approach).

The environments in which artificial moral agents will operate will not be fully predictable and controlled. And if we want them to be successful they will have to have the ability to determine their actions autonomously, that is their behaviour and action repertoire are not predictable (note that determinism does not imply predictability!). And in time new abilities will be added onto the existing set (dynamically). The agents will be active in different application domains and engage in both non-moral and moral activities. They will pursue different

goals at the same time, and have to make decisions within limited time and restricted information. The list of requirements following from these observations are:

1) mixed moral and on-moral goals support
2) mixed moral and on-moral activity support
3) bounded rationality, time & resource constraint
4) extendibility

This is not an exhaustive list of all requirements, but it does capture several key ones.

Following from these requirements a set of design principles can be formulated (from which I omit the software engineering oriented ones):

1) agents behave both pro-actively, goal driven and reactively
2) behavior is build from small action components that can be put to various uses and re-configured at run-time to form different behavior patterns
3) agents can decide if and when to update their information base
4) agents interact with each other and the environment

These principles are in support of what Korniek and Uzgalis observed as important characteristics of successful biological systems, which I feel are directly relevant in this context (Korniek and Uzgalis, 2002, 85).

- emergent behavior - not all behavior is specified upfront
- redundant degrees of freedom - more ways to achieve a particular goal
- no central director of action
- think local, act local - local scope of control

These are characteristics agents should have, and/or display in order to be able to meet to above requirements.

Building blocks: modelling & implementation

To model agent behaviour the belief-desire-intention model, BDI-model (Bratman, 1987) provides a good foundation. It captures both bounded rationality, and the goal oriented aspect that is required for autonomous agents. There are two important elements missing in the BDI-model to make it suitable for modelling artificial moral agents: the deontic element and the action element. Therefore the BDI-model is extended through the deontic-epistemic-action logic, framework, DEAL framework (Van den Hoven and Lokhorst, 2002). The deontic logic covers the deontic concepts of 'obligation', 'permission', and 'forbidden'. Epistemic logic expresses the things we know and belief. And the action logic

allows us to reason, through the STIT - see to it that ñ operator to reason about actions. Combined we can construct propositions like

$$(27) \quad B_i(G(\Phi)) \rightarrow O([i\ \mathrm{STIT}\ \Phi])$$

meaning if *i* believes that '*Φ*' is morally good than *i* should act in such as way that '*Φ*' is brought about

The modal logic in this approach is used as a specification language rather than as formal logic for theorem proving. My sentiment in this respect is very much that of (Halpern, 2000, 1)

> "McCarthy want robots to reason with a formal logic. I'm not sure that is either necessary or desirable. Humans certainly don't do much reasoning in a formal logic to help them in deciding how to act; it's not clear to me that robots need to either. Robots should use whatever formalisms help them deciding. Robots should use whatever formalisms help them make decisions. I do think logic is a useful tool for clarifying subtleties and for systems designers to reason about systems (e.g., for robot designers to reason about robots), but it's not clear to me that it's as useful for the robots themselves."

Of course, software agents and robots do need logic. It is inconceivable that they reason without logic. But they need not necessarily have a complete and sound logic system. Using the BDI and DEAL modelling tools agents can be attributed both moral and non-moral desires, have beliefs about what is morally obligatory or permissible, form multiple intentions, decide to act on them, and actually enact them.

The implementation of these models is done using the JACK agent language (JACK). This abstraction layer on top of the Java programming language provides support for multi-agent systems, based on the BDI-model. It provides components for agents, team, modalities of belief, desire and intention. Plans are sequences of actions that provide low level components for behaviour. Plans are connected at run-time to provide complex behaviour. To facilitate meta-level reasoning there are various mechanisms ranging from hard-wired, instantaneous choice to explicit reasoning about options and preferences. Beliefs are implemented as n-tuples (first-order relational models) that support both open-world and closed-world semantics. Plans can be reconsidered based on new desires (events), and new information becoming available. Plans can be linked to beliefs using logic variables for which agents can find multiple bindings. An intricate mechanism that allows to cater for multiple instances of an objective.

Some results

These building blocks proved to be a powerful basis to capture, model and implement aspects of moral agency, though still a great deal needs to be done (Wiegel 2006a. Below I list some examples of the (support for) insights gained.

1) Negative moral commands are different in nature from the way we talk about them. 'Thou shall not...' is a general form of moral command telling you what not to *do*. Trying to implement such obligations proved rather hard. The reason is that these commands are in fact not about acts as we talk about them: do not kill, do not lie, etc. There are many ways (in fact infinite) in which one can kill, and identifying each of them is impossible. These moral commands are actually about classes of action that are characterized by their outcome, e.g. bringing about a state of affairs in which someone does not have a beating hart, with all the qualifiers about intentionality, etc. This observation implies that agents must have an explicit conception (right or wrong) about the outcomes of their actions, and the ability to classify them accordingly.

2) Morality must act as both restraint and goal-director. Moral reasoning in artificial agents much function within a large context. An artificial *moral* agent in its own right does not have much practical relevance when it comes to application. An agent can have as one of its goals or desires to be a moral agent, but never as its only or primary goal. So the implementation of moral reasoning capability must always be in the context of some application in which it acts as a constraint on the other goals and action.

3) Moral decision making or decision making with moral constraints must take various restricting factors into account. One, an agent is open to permanent information feeds. It must be able to decide on ignoring that information or taking it into account. Two, once goals have been set, these goals must have a certain stickiness. Permanent goal revision would have a paralyzing effect on an agent and possibly prevent decision making. Three, different situations require different decision-making mechanisms. In some situations elaborate fact finding and deliberation are possible, in other the decision must be instantaneous.

All these three restrictions refer to resource boundedness and the time consuming dimension of decision making. To cater for these the BDI-model provides a good basis. The software implementation offers three decision making, or meta-level reasoning mechanisms: hardwired, ordinal and cardinal ranking and explicit reasoning.

4) Moral epistemology will prove to be the biggest challenge for the implementation of artificial moral agents. (Meta-)ethical theories make strong epistemological claims. E.g. if moral qualities are supervenient on natural qualities how does an agent perceive these natural qualities and deduce the moral dimension from them? If moral qualities are non-reducible how does an agent intuit or perceive them? How does an agent come to approve of something, and hence call it morally good?

These are very hard questions for which no clear answers are available that can be formalized. Moral epistemology of artificial agents will have to be strong typed for the near future. This means that perception, events, facts, etc. have to be typed at design-time. This means, for example, that events will need to have an attribute identifying its type, which then allows the agent to interpret it.

With all the promising results a note of reserve is called for. With Wooldridge, writing on his logic for rational agents, I would like to say (Wooldridge, 2000, 91)

> "Belief, desire and intention are in reality far too subtle, intricate and fuzzy to be captured completely in a logic [...] if such a theory was our goal, then the formalism would fail to satisfy it. However, the logic is emphatically not intended to serve as such a theory. Indeed, it seems that any theory which did fully capture all nuances of belief, desire and intention in humans would be of curiosity value only: it would in all likelihood be too complex and involved to be of much use for anything, let alone for building artificial agents."

This goes for logic, and for any other means to capture the aspects of moral reasoning. With our current understanding of moral reasoning, the challenges of moral epistemology and the state of technology development (far and impressive though it is, still falling short by a long way) we will have to settle for limited artificial moral agents. The explicit ethical agent is an ambitious goal though within the realm of the possible in our personal foreseeable future.

Nederlandse samenvatting

Er zijn tenminste drie perspectieven op het experimenteren met kunstmatige agenten en ethiek, die de combinatie buitengewoon interessant maken.
Ten eerste, door te trachten kunstmatige agenten te construeren die beschikken over moreel redeneervermogen, wordt ons begrip van moraliteit en moreel redeneren tot het uiterste uitgedaagd. Het betekent dat we ons zowel op het hoogste niveau rekenschap moeten geven over zaken als architectuur en structuren, als over de kleinste details. De computer is namelijk een instrument dat ons dwingt alle kennis die we bezitten te expliciteren. Zonder die expliciete instructie functioneert hij namelijk niet. Daarnaast wordt het formele begrippenapparaat waarin we onze kennis uitdrukken, de modale logica in dit geval, ten volle benut om de kennis zo te formaliseren dat ze door computerprogrammeurs kan worden begrepen en omgezet in computertaal.
Ten tweede, de complexiteit van de situaties waarover morele theorieën een uitspraak moeten doen, neemt alsmaar toe. Moderne samenlevingen met grote, multi-nationale organisaties en bedrijven die intensief gebruik maken van informatie-netwerken, vormen een steeds moeilijker te omvatten achtergrond waartegen morele theorieën uitspraken moeten doen. De filosofie-beoefening kan in deze context niet zonder gecomputeriseerde ondersteuning.
Ten derde, met de voortschrijdende technologische ontwikkelingen spelen artefacten in toenemende mate een rol in ons leven. Niet alleen bevatten ze informatie over ons, ze handelen ook namens ons, en beslissen over ons. Met de toenemende autonomie van de technologische constructies neemt ook onze behoefte toe om zeker te stellen dat het functioneren van deze artefacten in lijn is met wat wij van ze verwachten. We willen zeker stellen dat er geen misbruik gemaakt wordt, geen schade aan ons of anderen wordt gedaan, enz. Naast dit negatief ingestoken aspect is er ook een positieve invalshoek. Als we machines willen ontwikkelen die ons helpen, bijvoorbeeld in situaties waar mensen moeilijk uit de voeten kunnen, of het te gevaarlijk is, denk aan hulp in rampgebieden, dan hebben ze alle redeneervermogen nodig dat we ze kunnen leveren, inclusief moreel redeneervermogen.

Wanneer we trachten kunstmatige, morele agenten te construeren moeten we definiÎren waar we ons op richten. Moor stelt een classificatie van vier typen agenten voor: 'ethical impact agents', 'implicit ethical agents', 'explicit ethical agents' en 'full ethical agents' (Moor 2006). Deze classificatie loopt van simpele machines die een ethische uitwerking hebben eenvoudig vanwege hun bestaan

tot complete, mensgelijkende agenten met bewustzijn, en het vermogen om een plausibele verklaring te geven voor hun handelen.
Gegeven onze huidige stand van kennis over morele redeneren, kunstmatige intelligentie, epistemologische gereedschappen, enz., is het hoogst haalbare de ontwikkeling van expliciete, ethische agenten. Zij kunnen morele beslissingen nemen die niet hard gecodeerd zijn in hun ontwerp en hebben een zeker vermogen om rekenschap te geven over hun beslissingen.
Vanuit het perspectief van sofisticatie kunnen we nu spreken van een tweede generatie agenten. De eerste generatie was grensverleggend in de zin dat ze voor het eerst morele beslissingen implementeerde in een kunstmatige agent. Danielson (1992) is hiervan een paradigmatisch voorbeeld. Deze agenten zijn beperkt in hun ontwerp en uitvoering. De huidige generatie heeft een uitgebreider pallet van mogelijkheden tot, onder andere, teamwerk, meerdere doelen van morele en a-morele aard, en zekere vormen van epistemologische uitrusting. Een volgende generatie zal waarschijnlijk een fysieke verschijningsvorm hebben met sterk verbeterde epistemologische vaardigheden.

Filosofen verlaten meer en meer de studeerkamer (armchair), en houden zich bezig met het testen van hun theorieën, het verzamelen van gegevens met behulp van computers, en met het ontwikkelen van computermodellen. De introductie van experimenten in de filosofie heeft een heel nieuw werkterrein blootgelegd. Er zijn vele initiatieven die met behulp van verschillende technieken en hulpmiddelen filosofische theorieën testen. Niettemin is de claim in dit onderzoek dat vele van deze onderzoeken eenzelfde aanpak volgen. In dit onderzoek is het methodologische raamwerk geschetst dat gedeeld wordt door verschillende onderzoekers op dit nieuwe terrein. Danielson en Thagard zijn twee paradigmatische voorbeelden van onderzoekers die met behulp van computermodellen hun theorieën onderzoeken.
In de kern is experimenteren een vertaalproces: van de theorie naar de techniek waarin wordt geëxperimenteerd. Dit vertaalproces verloopt in negen stappen.
1) De theorie wordt opgebroken in assumpties, premissen, mechanismen, voorspellingen, enz.,
2) die worden vertaald naar een concrete, experimentele toepassing (en geen abstracties blijven)
3) en die worden vertaald naar een tussenliggend, conceptuele raamwerk.
4) De concrete experimentele toepassing wordt ook vertaald naar de begrippen van het conceptuele raamwerk.
5) De theorie wordt geïmplementeerd in het laboratorium gebaseerd op de eisen van het tussenliggende conceptuele raamwerk
6) zodanig dat het experiment er in weergegeven wordt.
7) De experimenten worden uitgevoerd

8) en de resultaten worden terugvertaald naar de begrippen uit de theorie.
9) De theorie kan vervolgens worden verworpen, bevestigd of aangepast.

Bij het experimenteren in de filosofie moet men rekening houden met een aantal methodologische kwesties. Deze keuze voor de te gebruiken technieken en het conceptuele raamwerk kunnen de uitkomst van de experimenten beïnvloeden. Standaards en protocollen voor het uitvoeren van experimenten zijn nog niet uitgewerkt.

In het kader van dit onderzoek zijn drie verschillende experimenten opgezet en uitgevoerd. Er is een experiment gedaan met betrekking tot de notie van 'moreel gebod'. Wat is ons begrip van deze notie, en verandert dit begrip als het in een experimentele context wordt toegepast. Een ander experiment richt zich op een lang lopend debat rondom regel- en daadutilisme. Twee verschillende varianten die het morele goede baseren op het nut. Via experimenten is getracht extra inzicht te verschaffen in de argumenten voor en tegen beide varianten. Het derde experiment betreft de bescherming van privacy. In een wereld waar informatie tussen veel verschillende mensen wordt gedeeld is de vraag hoe de privacy kan worden gewaarborgd.
De experimenten worden opgezet met behulp van modale logica en multi-agent software systemen. Het construeren van de experimenten gebeurt in de stappen: modelleren, ontwerpen en coderen. Voor stap een wordt gebruikt gemaakt van DEAL, deontic epistemic action logic (Van den Hoven and Lokhorst, 2002), tezamen met het 'belief desire intention'-model (BDI-model, Bratman, 1987). Beide modellen zijn gebaseerd op modale logica. Ze bieden een raamwerk om het vereist gedrag te specificeren dat preciezer is dan onze omgangstaal.
Voor het ontwerp en de implementatie wordt gebruikt gemaakt van de JACK ontwikkelomgeving. In die ontwikkelomgeving zijn onder meer de volgende componenten voor handen.

- Beliefsets - 'beliefs' representeren het epistemische aspect
- Events - goals' en 'desires', zijn type van events die het doelgerichte gedrag vorm geven
- Actions, plans and reasoning methods - zij implementeren de componenten van actie en intentie
- Agents - zijn de containers waarin de andere componenten worden samengebracht
- De Java programmeertaal waarin de extensies op het standaard raamwerk kunnen worden geprogrammeerd

Wat is er nodig om een kunstmatige agent uit te rusten met moreel redeneervermogen? Deze vraag wordt nader onderzocht aan de hand van de vraag hoe een moreel gebod geïmplementeerd kan worden.
Via het opzetten van experimenten is gebleken dat het gangbare gebruik van morele geboden de onderliggende structuur verhult. Deze structuur is niet gericht op handelingen ('gij zult niet doden') maar op de uitkomsten van die handelingen ('gij zult geen toestand creëren waarin iemand geen hartslag meer heeft', of wat er medisch geldt als dood. Het experiment toont daarmee dat het mogelijk is om morele noties te implementeren, maar tevens hoe 'vaag' onze noties vaak nog zijn. Bovendien blijkt dat morele theorieën zware aannames maken ten aanzien van de morele epistemologie. De implementatie daarvan in software zal nog een harde dobber blijken te zijn.
Een ander experiment gaat over de vraag welke rol informatie, en de kosten van informatie, spelen in de effectiviteit van regel- en daadutilisme. Uit het uitvoeren van experimenten is niet alleen gebleken dat informatie een belangrijke rol speelt maar tevens de volgende elementen: de groepsgrootte; besluitvorming algoritmes waarin onzekerheid een rol speelt; de inclinatie tot bepaald gedrag (sociale mores, cultuur). In het lopende debat over regel- en daadutilisme kan worden geconcludeerd dat de vaak scherpe tegenstelling niet zo scherp behoeft te zijn. Beide vormen eerder twee verschillende posities op een continuüm (ze zijn functioneel equivalent), afhankelijk van de precieze aannames omtrent de beschikbaarheid van informatie. Tevens blijkt dat regelutilisme egalitairder is dan daadutilisme. Als de groepen groter worden waarin de agenten opereren, neemt de effectiviteit van het daadutilisme toe. Bij een toenemende invloed van instituties blijk het daadutilisme ook beter te functioneren. Dit is de paradox dat daar waar het regelutilisme het meeste effect heeft gehad het daadutlisme beter functioneert. De verklaring ligt in het gegeven dat met de toename van instituties (regels) de kosten van informatie dalen. Hetgeen de daadutilisten beter doet functioneren.
In het experiment rondom privacy is een toepassing binnen de verzekeringindustrie gekozen. In het verzekeringsbedrijf wordt informatie van en over klanten verwerkt van zakelijke, prive en medische aard. Er zijn veel partijen betrokken (bijvoorbeeld artsen, bloedbanken en notarissen). Bij de verwerking en uitwisseling van die informatie kan de privacy in het geding komen. Er zijn veel partijen en praktisch gesproken kan het moeilijk zijn om te waarborgen dat iedere betrokkene (inclusief de systemen) zich houdt aan alle richtlijnen met betrekking tot bescherming van de privacy.
Uit het experiment blijkt dat het zowel mogelijk is als nuttig om de vereisten rondom privacy te formaliseren met behulp van BDI en DEAL. De uitvoering in de software van JACK blijkt goed te functioneren. Daarmee is de deur geopend

naar bredere toepassingen in, bijvoorbeeld, patiëntendossiers en financiële adviesbedrijven waar de zogenaamde 'chinese walls' tussen de adviseringstak en de accountancy-tak geborgd dienen te worden.

De implementatie van DEAL en BDI blijkt goed mogelijk. De omzetting ervan in uitvoerbare computercode gaat goed vanwege de hoge formaliseringsgraad. Ethische begrippen laten zich goed omzetten in de logica en de software. Beide blijken expressief genoeg te zijn. De experimenten dragen bij aan het betere begrip van morele theorieën. Tevens is het makkelijker om de consequenties van een theorie in te schatten, met name in situaties waar de complexiteit groot is.

Tenslotte blijkt de toepassingsmogelijkheid voor maatschappelijk relevante vraagstukken zoals privacy groot te zijn. Dit neemt niet weg dat er nog veel werk te verrichten is, ondermeer op het gebied van 'lerende agenten', teamwerk en morele epistemologie.

Index

Simon Stevin (1548-1620)

'Wonder en is gheen Wonder'

This series in the philosophy of technology is named after the Dutch / Flemish natural philosopher, scientist and engineer Simon Stevin. He was an extraordinary versatile person. He published, among other things, on arithmetic, accounting, geometry, mechanics, hydrostatics, astronomy, theory of measurement, civil engineering, the theory of music, and civil citizenship. He wrote the very first treatise on logic in Dutch, which he considered to be a superior language for scientific purposes. The relation between theory and practice is a main topic in his work. In addition to his theoretical publications, he held a large number of patents, and was actively involved as an engineer in the building of windmills, harbours, and fortifications for the Dutch prince Maurits. He is famous for having constructed large sailing carriages.

Little is known about his personal life. He was probably born in 1548 in Bruges (Flanders) and went to Leiden in 1581, where he took up his studies at the university two years later. His work was published between 1581 and 1617. He was an early defender of the Copernican worldview, which did not make him popular in religious circles. He died in 1620, but the exact date and the place of his burial are unknown. Philosophically he was a pragmatic rationalist for whom every phenomenon, however mysterious, ultimately had a scientific explanation. Hence his dictum 'Wonder is no Wonder', which he used on the cover of several of his own books.

In this book, the extend to which we can equip artificial agents with moral reasoning capacity is investigated. Attempting to create artificial agents with moral reasoning capabilities challenges our understanding of morality and moral reasoning to its utmost. It also helps philosophers dealing with the inherent complexity of modern organizations. Modern society with large multi-national organizations and extensive information infrastructures provides a backdrop for moral theories that is hard to encompass through mere theorising. Computerized support for theorising is needed to be able to fully grasp and address the inherent complexity. Using moral reasoning capacity will help us addressing the challenges that technological artefacts pose. They do not only contain information about us, they start to act on our behalves.With the increasing autonomy comes an increased need to ensure that their behaviour is in line with what we expect from them. To investigate and address these issues a laboratoy for philosophy is outlined: SophoLab. It consists of a methodology; a framework of modal logic, DEAL; and multi-agent software systems. SophoLab provides the basis for an experimental, computational philosophy. Its viability and usefulness are demonstrated through several experiments.

‘Wonder en is gheen wonder’

TU/e technische universiteit eindhoven

TU Delft
Technische Universiteit Delft