Distributed Opportunistic Argumentation Guided by Autonomous Agent Interaction

Paul William Martin

Doctor of Philosophy

Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh

2011

Abstract

Within a distributed system, autonomous agents may find it necessary to cooperate in order to achieve their objectives. Interaction protocols provide standard frameworks within which to conduct common classes of interaction, but they are only useful when the agents using them have a common interpretation of the constraints imposed by those protocols. In *open* systems, where there are no system-wide objectives and components are contributed from a variety of sources, this is difficult to ensure.

An agent within a sufficiently complex environment will find it necessary to draw inferences from information sources of varying integrity and completeness. Given flawed or incomplete information, it may be necessary for an agent to resort to non-monotonic reasoning in order to be able to make concrete decisions within limited windows of opportunity. This can be expected to create inconsistencies in the joint beliefs of agents which can only be repaired by dialogue between peers. To verify and repair all possible sources of inconsistency is impractical for any sizable body of inference however — any belief revision must therefore be subject to prioritisation.

In this thesis, we introduce a mechanism by which agents can perform opportunistic argumentation during dialogue in order to perform distributed belief revision. An *interaction portrayal* uses the protocol for a given interaction to identify the logical constraints which must be resolved during the interaction as it unfolds. It then compares and reconciles the expectations of agents prior to the resolution of those constraints by generating and maintaining a system of arguments. The composition and scope of arguments is restricted in order to minimise the information exchange whilst still trying to ensure that all available admissible viewpoints are adequately represented immediately prior to any decision. This serves both to make interaction more robust (by allowing agents to make decisions based on the distributed wisdom of its peer group without being explicitly directed by a protocol) and to reconcile beliefs in a prioritised fashion (by focusing only on those beliefs which directly influence the outcome of an interaction as determined by its protocol).

Acknowledgements

Writing a thesis has been an *interesting* experience, for want of a better term. It has at times been slow and frustrating, and at other times been almost pleasant. In any case, it has been a worthwhile endeavour, if only because I learned a lot about the challenges of trying to distill one's thoughts onto the written (or printed) page. But enough of my musings …

I would like to thank my supervisors, David Robertson and Michael Rovatsos, whose patience and good advice has been invaluable, even if it appeared at times that I wasn't quite listening.

I would like to thank all of my contemporaries in CISA who have been and gone, and those who are still around, particularly those who have shared offices with me in both the Tower and the Forum; you know who you are and it would have been boring without you around.

I would like (grudgingly) to thank my friend and contemporary Tom Clayton, who insisted in giving me advice even when it wasn't wanted. It might have helped — I concede nothing.

Finally, I would like to thank my family, who encouraged me throughout my studies despite having no idea what it was I did.

Thanks.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Paul William Martin*)

Table of Contents

Chapter 1

Introduction

Distributed artificial intelligence concerns itself with the deployment and coordination of multi-agent systems. In particular, it is concerned with the achievement of complex collaborative behaviours with a minimum of scripting and oversight. Ideally, it should be possible to assign some task to a system without dictating how that task should be performed, such that the system will then spontaneously assemble itself into a configuration capable of efficiently executing that task.

Such assembly requires that the agents constituting a system be able to collectively and autonomously choreograph themselves in order to perform any number of arbitrary tasks. This in turn requires shared frameworks for communication and coordination. It also requires that agents be able to reason about their own private states and their external surroundings so that they can play their roles effectively within such frameworks. Given a sufficiently complex environment however, agents may need to reason with incomplete knowledge and make assumptions based on unproven hypotheses where no evidence exists to the contrary. This thesis concerns itself with the conduct of interactions between autonomous yet truthful agents in knowledge-rich environments, and with how the chosen beliefs of agents influence such interactions. It asks under what circumstances it is possible to control and direct such influence in order to achieve outcomes which best reflect the combined wisdom of all involved agents.

A basic assumption on the part of this thesis is that it is desirable for models of interaction to avoid pedantry and over-reliance on 'ritual behaviours' — behaviours which are indecipherable to outsiders, particularly potential participants. In other words, any given interaction protocol should be broadly applicable to a given class of problem and any action demanded by such a protocol should clearly lead towards one of a finite number of clear outcomes. Given our desire that models should be both

lightweight and generic, it then becomes evident that agents cannot be totally subservient to any model, because so many unstated factors are left to the discretion of the individual acting in its given role. Discretion does not entail isolation however. There is often no reason why agents cannot freely solicit the opinions and expectations of their peers during an interaction in order to better discharge their responsibilities. This can be done in harmony with, rather than in spite of, the interaction protocols to which they must adhere.

Another assumption made in this thesis is that agents in a multi-agent system do not exist in a state of hermetic seclusion. Instead, they are continuously engaged with their environment. However whilst an agent may have at its disposal a great volume of information, that information is not always of the greatest integrity. It may in fact contain several inconsistencies between and even within specific information sources, usually introduced by mistaken assumptions. Agents therefore must have the ability to evaluate any article of information according to their needs, and ultimately disseminate and test their conclusions — this can easily be done during dialogue with their peers, when such conclusions may shape the course of an interaction. From this perspective, interaction between agents can be seen as serving as an engine for distributed belief maintenance.

With these ideas in mind, this thesis introduces a mechanism by which arbitrary interactions based on shared interaction protocols can be augmented to allow the dissemination of useful information whilst providing an opportunity to resolve the inevitable conflicts which arise from otherwise honest agents having contradictory beliefs. By engaging in an opportunistic process of distributed argumentation, a vessel into which agents can articulate certain expectations and beliefs provenant to the interaction, called here a *portrayal*, can be created which frees the accompanying interaction protocol to concern itself solely with the choreography of the interaction itself, rather than with the minutiae of how constraints on such choreography are best resolved. We use this mechanism to demonstrate the synchronisation of agent beliefs within a restricted argument space — a state in which the distributed wisdom of agents as it portends to the interaction at hand can be leveraged with minimal exchange of information. It will be shown that such synchronisation allows both for more robust interactions and for the unsolicited propagation of common theories within an agent population, which provides a stronger basis for further interactions between a given group of agents.

Let us begin by describing the motivating problem.

1.1 Motivation

This thesis was originally motivated by a desire to make interactions (specifically, coordinated exchanges of data and services between autonomous processes) more reliable; a study was made of interactions conducted using distributed dialogue protocols written in the *Lightweight Coordination Calculus* (or LCC) [Robertson, 2004]. Motivated itself by a desire for a decentralised approach to the *electronic institutions* model of agent interaction (E-Institutions) [Esteva et al., 2001], LCC protocols identify the abstract roles played by agents in a given class of interaction, and define an independent process model for each. Agents assume roles as necessary, voluntarily limiting their behaviour to fit with the constraints of the given process model, and their actions are then coordinated with those of their peers by means of message exchange. Given common knowledge of the protocol itself, any message received from a peer can then be taken as a commitment to the effect that any prerequisite decisions which must be made prior to sending such a message have been made, and any consequent actions demanded by its role model will be taken. Consequently, there is (for many interactions at least) no need for any central oversight over interaction.

Nonetheless, it became clear that collaboration between agents is heavily influenced by the assumptions each agent brings into an interaction. In LCC, progress towards any particular outcome is determined by the evaluation of certain logical constraints, which individual peers[1] are free to decide based on their own personal theories. Since how an agent interprets its environment influences the decisions it makes, it follows that ontological, epistemological and historical concerns will all affect the outcome of any task it engages in. Simply put, an agent could behave in a manner entirely unexpected by its peers by simple virtue of evaluating a given constraint in an unexpected way. Of particular concern were cases in which an agent would *appear* to behave consistently with expectations, but where there existed some hidden discrepancy between its reasoning and that of its peers which would lead to violation of some social contract at a later time (perhaps not even during the same interaction).

Example 1.1 *A typical interaction would be one in which an agent engages with its peers in order to bring about an action it could not perform by itself. Assume that there exists an LCC protocol which allows an agent to acquire access to a privileged resource by first acquiring the support of a trusted peer.*[2] *Such a protocol would define*

[1] Throughout this thesis, we use the term 'peer' to mean an autonomous agent, rather than a simple distributed process.

[2] Indeed there does exist such a protocol, in Chapter 6.

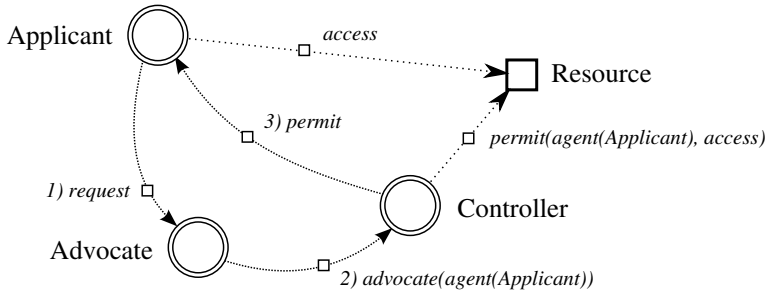


Figure 1.1: *A simple interaction in which an agent engages with its peers in order to gain access to some resource.*

three roles:

Applicant — *An agent acting in role* applicant *requests the advocacy of an existing patron of an inaccessible resource. If the advocate accepts the applicant's request and the resource's controller accepts the advocate's recommendation, then success is confirmed upon being able to access the resource. This role specifies two constraints (one of which,* accessible*, is checked both at the start and end of interaction):*

accessible(*Applicant*, *Resource*) — *The applicant must not initially be able to access the desired resource, but should have access after receiving permission from the controller.*

patron(*Advocate*, *Resource*) — *The applicant must be able to identify a suitable patron of that resource to be its advocate.*

Advocate — *The agent designated the* advocate *will present an applicant's case to the resource's controller should it consider the applicant trustworthy. Thus* advocate *will only proceed if the following constraints are satisfied:*

controller(*Controller*, *Resource*) — *The advocate must be able to identify an agent capable of deciding peers' access to the resource.*

trustworthy(*Applicant*, *Resource*) — *The advocate must decide whether or not the applicant should be trusted with access to the resource.*

Controller — *The* controller *permits access to a resource only if an applicant is both eligible and has been advocated by a peer it already trusts. Thus the following constraints are imposed:*

trusts(*Controller*, *Advocate*) — *The controller must already trust the advocate, and thus its judgement.*

eligible(*Applicant*, *Resource*) — *The controller must consider whether or not the applicant is even eligible for access.*

Each agent may possess its own conception of what the 'correct' interpretation of a given proposition might be. In this case, most propositions can be interpreted rather subjectively. For example, the evaluation of trust can be based on reputation, observations of past behaviour, the possession of credentials, or the perceived intent of the subject; it may also be universal, or circumstantial. This protocol also relies on agents being able to distinguish between trusting *a peer and considering a peer to be* trustworthy[3]. *Eligibility may be based on qualities of the agent, the logistics of resource provision, or both. It also cannot be ascertained from the protocol itself exactly what constitutes a 'patron' of a resource.*

In the above example, an interaction could go awry if the advocate declares the applicant trustworthy, but the applicant later abuses that trust. The problem of concern was not merely one of trust however. A similar issue would arise if the controller decided that the applicant was eligible based on false assumptions about the applicant. This reveals a related problem — that the satisfaction of certain constraints in an LCC protocol is often reliant on information presumably gathered prior to interaction. If the given protocol does not provide an explicit mechanism by which an agent can solicit information from its peers, then many propositions will only be resolvable if the agent already knows enough to make a decision based on whatever interpretation of that proposition it applies, or possesses the initiative (and privilege) to retrieve the information in the midst of an active interaction. The only other option is to make an unverified assumption, which could lead to an unjustified interaction outcome.

It is worth immediately noting that, to a certain extent, any constraint interpretation problem can be resolved by the prior construction of more rigorous protocols. Ambiguous constraints can be decomposed into a set of simpler, more objective propositions. Likewise, for any decision that may require additional testimony from peers, the protocol can explicitly provide a process by which an agent can query those peers, and then collate the results. In doing this however, an interaction protocol (whether written in LCC or otherwise) is likely to become less generically applicable to different variations upon the same basic task. Such protocols tend to become useful only

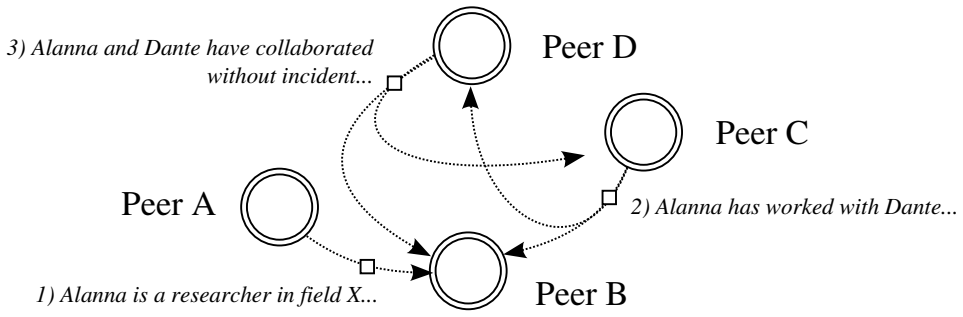
[3] Something which seems to regularly confuse otherwise intelligent people all the time.

in the circumstances specifically envisaged by the protocol designer — they demand that a given constraint is resolved in a certain way or they demand that information is gathered by a particular routine. It was quickly decided that if LCC was to be useful in arbitary systems (such as those which allow free contribution of peers and services, the very kind of system for which LCC was designed), then its protocols needed to be kept flexible and concise, and individual agents needed to be afforded the discretion to make decisions in whatever manner they deemed most appropriate.

Thus it became of great interest to investigate the possibility of a mechanism which would allow agents to generically explore constraints imposed on an interaction; a mechanism which could operate alongside any interaction protocol, preferably without any need to modify existing protocols. Such a thing would be useful in a number of scenarios, all of which can naturally arise during archetypal interactions. For example:

Soliciting advice from peers — An agent may have to decide between a number of actions based on its evaluation of a certain unknown quantity. The agent may be able to make a more informed decision however, if it is first allowed to solicit advice from any of its peers.

Example 1.2 *Benjamin has to decide whether or not Alanna is trustworthy in a given role. Not knowing much about Alanna, Benjamin would preferably want more information about her and her past interactions with other peers. Thus Benjamin invites testimony from his peers (including Alanna):*

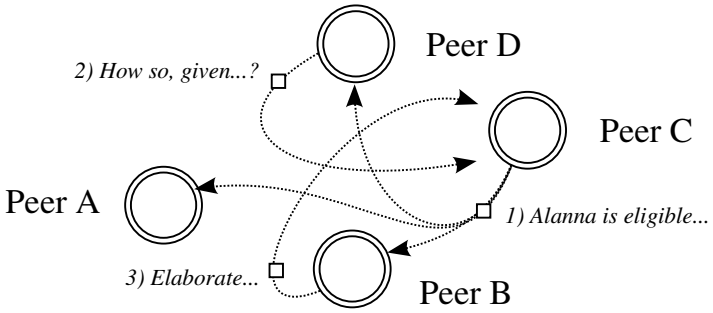


Alanna volunteers her credentials (in this case, that she is a researcher in a field relevant to the given role, with the implication that this makes her suitable for the role), whilst Charlotte, with Dante's assistance, provides reference to an analogous case in which Alanna justified the trust given to her.

This is an example of a *dissemination* dialogue, where missing data is solicited or volunteered in order to apply some decision procedure.

Verifying decisions — A particular agent may be tasked with making decisions for its peer group. There may however be cases where the agent would make a decision not admissible to its peers. It would be useful if peers were able to query the agent's decisions on their own initiative, but without committing to then systematically verifying *every* decision taken.

Example 1.3 *Charlotte has determined that Alanna is eligible for access to a given resource; normally this is a decision which peers are happy to delegate responsibility for:*

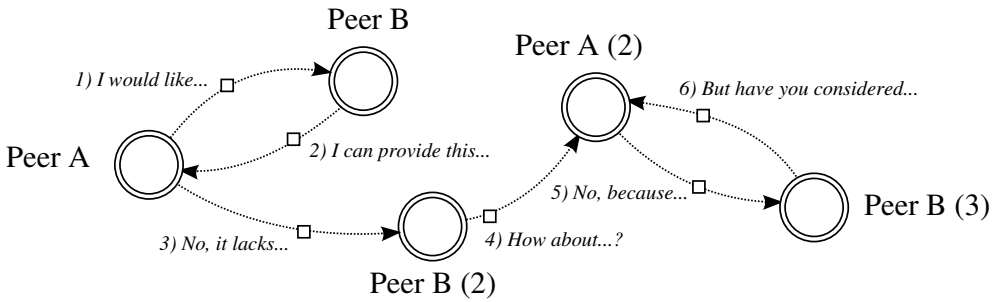


Dante however, perhaps having assumed that Alanna was not eligible, queries how Charlotte came to her decision. Similarly, Benjamin may be curious as to how Charlotte makes such decisions, having an incomplete understanding of the underlying problem.

This is an example of an *investigation* dialogue, where requests are made for statements to be elaborated upon in order to allow other agents to verify the reasoning of peers.

Negotiating services — Two (or more) agents may need to negotiate the provision of some service. Where the inherent properties of a service are not enshrined within an interaction protocol, the best way to handle negotiation may be to focus on the satisfaction of abstract constraints, which the peers can dispute until a mutually admissible contract is produced.

Example 1.4 *Alanna seeks a service which Benjamin provides. A successful interaction in this case is contingent on the seller being able to provide a specific product which matches the buyer's needs, and the buyer being willing to spend whatever resource is necessary to obtain the product offered:*

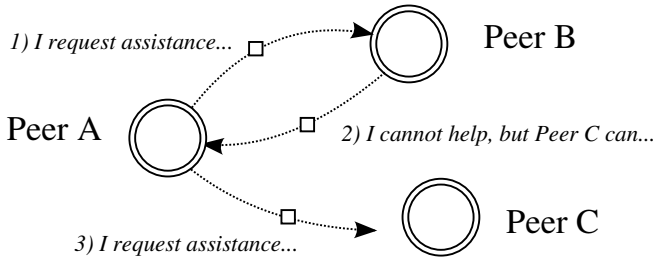


In this case, dialogue is used to describe the vital attributes of the product, explaining why they do or do not fulfill either constraint, inviting counter-points.

This is an example of a *negotiation* dialogue, where agents argue the merits of some transaction until consensus is reached as to its particulars or it is determined that the requirements of all parties cannot be reconciled.

Identifying alternative approaches — If an agent finds that their plans have gone awry (or are about to), it may be that its peers are able to provide information identifying alternative approaches, allowing the agent to salvage a failing interaction or re-attempt it under different circumstances.

Example 1.5 *Alanna would like Benjamin to assist it in some task, but Benjamin is unsuitable for the role Alanna has given him:*

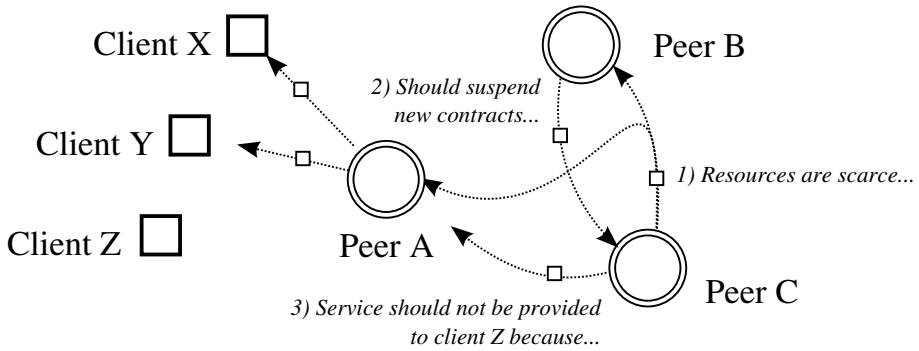


It should be possible for Benjamin to direct Alanna to another peer (in this case Charlotte) who is more suitable, allowing Alanna to repeat her request for assistance.

This is an example of an *exploration* dialogue, where agents collaborate to find new resolutions for constraints on interaction.

Identifying changes of state — If an agent observes a change in the environment in the midst of an interaction, one which has bearing on an active interaction, it would be helpful if that agent was able to convey that observation to its peers prior to the interaction proceeding further.

Example 1.6 *Alanna has been charged by her peers with determining whether or not providing a collective service to a prospective client is feasible:*



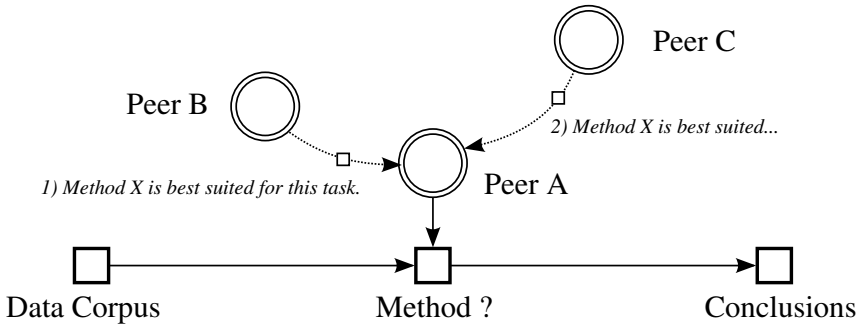
Benjamin determines that new contracts should be suspended however based on information disseminated by Charlotte; Charlotte then informs Alanna that a lack of resources will prevent future services from being fulfilled, stopping Alanna from offering the service until the situation is resolved.

This is an example of a *transition* dialogue, where the accepted situation given any earlier dialogue is changed to better reflect new information.

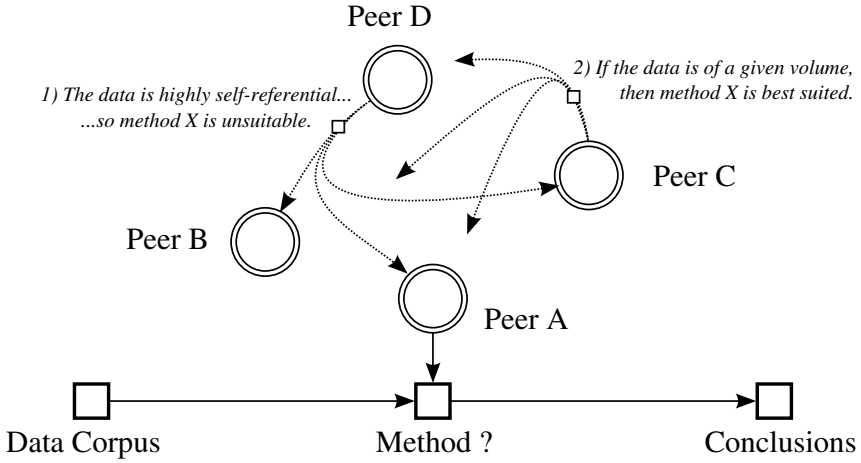
To be able to generate the kind of dialogues just described however requires there to be a standard process by which dialogue can be constructed without recourse to any data not found in the interaction state. There was already work done on using dialogue games to explicate elements of LCC protocols in [McGinnis et al., 2005], but in this case the focus would be on providing a generic problem to solve by agents during interaction which would result in the generation of the desired dialogues.

Such a decision problem would need to be sufficiently lightweight such that it would not unduly encumber any interaction it was attached to. In particular, in cases where additional dialogue would *not* be necessary to ensure an optimal interaction, the problem should be resolved all but immediately; where deemed by peers to be beneficial however, agents should be able to engage in more in-depth discussion.

Example 1.7 *Alanna has to choose one of a number of methods for interpreting some corpus of data, each with its own intrinsic qualities. Alanna can solicit advice from her peers, in this case Benjamin and Charlotte. If both peers are in agreement about which procedure to use, then whilst there is the possibility that they favour the same procedure for entirely contrary reasons, in general it is likely that selecting that procedure is a good plan that requires no further deliberation.*



On the other hand, if another peer Dante favours an alternative procedure, it is probably worth finding out why, as it is possible that Dante is in possession of important information not available to his peers.



Dante's input may lead to a more thorough discussion of the qualities of various procedures. Note that even if Dante is found to be misinformed, an opportunity has been given for any false assumptions on his part to be identified and purged.

It was found that the underlying decision problem could be understood as one of *disputation*. All the example scenarios identified could be understood as a consequence of aggressively positing some claim and inviting dissenting opinions. Where no dissent exists, there would be no further dialogue — otherwise, a group of agents could collectively generate a succession of assertions and arguments which would explore disputes and reveal essential information about the domain of discussion.

For example, looking at the different (overlapping) types of dialogue identified previously:

Dissemination — A dissemination dialogue can be invoked by positing hypotheses and inviting peers to respond. Agents can provide support for a given claim, or can provide information which undermines it.

> *I posit that Alanna is either trustworthy, or not. You posit that Alanna is trustworthy given some proposition. Another peer may posit that proposition as truth, or provide evidence to the contrary.*

Most dialogues will involve some degree of information dissemination, sometimes unsolicited.

> *You dispute another peer's claim, providing evidence to support your case. I observe your arguments, and learn something useful which I did not expect to learn in this context.*

Investigation — Given a claim by a peer, an agent can request that the peer expand upon its underlying reasoning, or can articulate its own explanation in order to provoke agreement or an alternative explanation from the peer.

> *I take issue with one of your claims, suggesting that you had made certain assumptions, which I reveal to be flawed. You admit my argument, but point out that you had inferred your conclusion from a different line of reasoning entirely, which you lay out for me.*

Negotiation — A sceptical agent can assume that a given artefact is *not* adequate, and force its peers to disprove each of its objections in turn until the final artefact meets its requirements.

> *I wish for you to perform for me a service. You make an offer, but I am sceptical as to whether it adequately fulfill my needs and so posit that it does not. You then provide arguments explaining to me why this is not the case, or if you cannot, you make me a different offer.*

Exploration — If every peer is allowed to posit possible resolutions to a given constraint, and is free to attack resolutions it finds unacceptable, then one can expect that any claims still admissible after dialogue has played out are still (to the best of the peer group's knowledge) viable possibilities.

> *We argue that certain peers should or should not be entrusted with some important task. Even should we fixate on a particular peer, our prior conclusions still stand, and we can return to them should our preferred choice be unwilling (or unable) to perform our task.*

Transition — Provided that the results of dialogue are persistent (at least over the course of an interaction), the admissibility of claims can change given new information.

> *We decide that Alanna is eligible for some privilege, based on the testimony given and the arguments made. At some later point, circumstances change, undermining a pivotal argument in her favour. By revisiting that argument, we can note the change, and re-evaluate Alanna's eligibility based on the arguments remaining.*

In such a manner can a hypothetical system of arguments be generated which explores the distributed knowledge of peers engaged in an interaction, but only where it is expected to influence the interaction. Such behaviour can be implemented by using assumption-based argumentation [Bondarenko et al., 1993], which can act as a generic proxy for many different forms of defeasible reasoning [Kakas and Toni, 1999].

Thus, we have our motivation — a desire to specify a distributed decision problem and a mechanism for solving that problem which will induce ad-hoc dialogue between agents, dialogue which will serve to inform the beliefs of agents such that more robust interactions can be conducted without resort to over-engineered protocols.

With this motivation in mind, consider now our contribution.

1.2 Contribution

In this thesis we specify a wholly distributed mechanism for opportunistic argumentation which allows *sincere* agents engaged in some interaction to share insights and reconcile conflicts of belief prior to making decisions which influence that interaction's outcome. We demonstrate that it is possible to interleave this mechanism with an ongoing cooperative agent interaction in order to make it more robust, to prevent ill-informed decisions from being made by individual agents and to ensure an outcome which better reflects the true state of the environment, as well as provide an improved basis for further interactions between those agents involved. We verify this mechanism by proof and validate it by example.

A *portrayal* is an annotated system of arguments articulated by agents during an interaction in support of, or in opposition to, particular resolutions of constraints imposed on interaction by an interaction protocol. It affords the agent tasked with deciding a given constraint the ability to forewarn its peers of the decision it would make in the current circumstances prior to actual resolution, inviting its peers to make arguments challenging that decision. It also allows agents acting in other roles to posit the decisions they would make if acting in the constrained role, permitting the deciding agent to inquire into their reasoning, though ultimate authority to determine a constraint re-

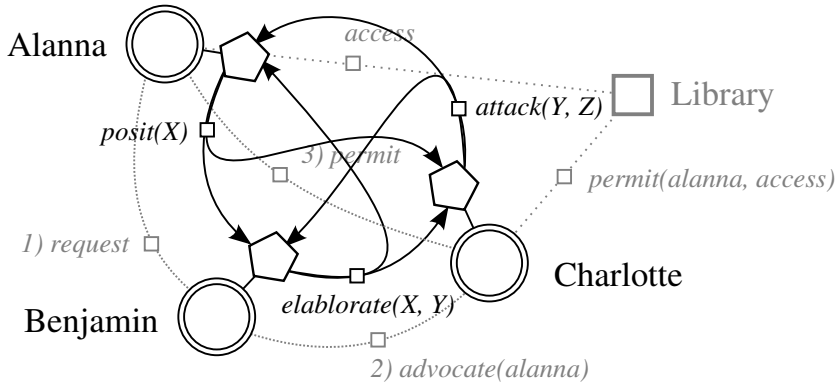


Figure 1.2: *An interaction between three agents can be augmented by constructing an interaction portrayal in tandem with the execution of a protocol.*

mains with that agent. The arguments and counter-arguments introduce new concepts to a group of peers and provide an opportunity to test existing assumptions. By provoking a mutual re-evaluation of beliefs, common theories are constructed supporting more robust interaction without infringing on the autonomy of individual agents.

Example 1.8 *Consider an interaction executed according to the protocol summarised in Example 1.1 wherein an agent Alanna takes on the role of* applicant *to a resource* library *and an agent Benjamin takes on the role of* advocate*. Over the course of dialogue, Alanna and Benjamin will express arguments supporting or refuting different resolutions of the constraints on interaction. For example:*[4]

Alanna: *I can be trusted with the library because I am a researcher in the field and have no reason to abuse your trust.*

Benjamin: *You are not trustworthy because you have violated trust in the past.*

Benjamin can further dispute Alanna's claim:

Benjamin: *You cannot be a researcher, because you do not appear to have a research topic.*

Alanna: *Actually, I am a researcher (presents credentials).*

Whilst likewise, Alanna can dispute Benjamin's:

Alanna: *There is no evidence that I have violated trust in the manner you suggest.*

Benjamin: *You are not trustworthy* specifically *because you abused access to* laboratory *(an analoguous resource).*

Alanna: *My prior point stands.*

[4]This example is drawn from the material in Chapter 6.

Benjamin, being assigned the task of evaluating trustworthy(alanna, library) *in his role as* advocate*, will then use the arguments made to decide whether or not Alanna is trustworthy subject to his own scepticism and bias.*

Our concern is not however with some intractable notion of perfect argumentation between agents which always provides optimal results but which is too cumbersome to deploy in practice, but with providing a 'good enough' service which *might* benefit the agents in an interaction and will not act to their detriment — efficiency is vital, insofar as portrayals must be something which can be deployed alongside any number of interactions without significant impact on the efficacy of a multi-agent system. This has required us to explore the notion of *argument spaces*, which serve to define the scope of argumentation in a heterogeneous reasoning environment by defining both relevance and necessary level of detail for arguments. By providing the means for agents to determine and, if necessary, refine the argument space of an interaction incrementally based on prior dialogue, we can ensure that portrayals are kept initially minimal, to be expanded only when agents decide that certain arguments must be explored further. Necessarily however, this forces us to consider the notion of a *good* argument space, one in which agents are still able to adequately explore the various facets of the problem at hand — defined as one in which all conclusions admissible given the distributed knowledge of agents can be seen to be admissible given a complete exploration of the argument space. Such a space can be seen as the 'goal state' of the decision problem which a portrayal is generated to solve. We identify the core properties of a 'good' portrayal argument space for an interaction, and we ensure that our mechanism produces portrayals which occupy that space, 'solving' the decision problem.

Example 1.9 *In the previous example, arguments are presented at a level of detail necessary to describe the essential dispute. However within the confines of its own knowledge base, an agent will reason at an arbitrary level of complexity commensurate with its knowledge of the domain. For example, Alanna might reason the following way:*

Alanna: I can be trusted with the library because:

- *I am a lecturer at the University of Edinburgh.*
- *Edinburgh is a recognised higher education institution.*
- *Any lecturer employed by a higher eduction institution is a researcher.*
- *I need access to certain data.*
- *That data is only found in* library.
- *There is no benefit in abusing access to the only source of needed data.*

– *A researcher with no incentive to abuse a resource can be trusted with that resource.*

This is much more detailed than is probably necessary to persuade Benjamin. In particular, each additional detail confers the possibility of further dispute. However, if the end conclusions are the same then such dispute will have no effect on the practical outcome of the interaction which argumentation is ostensibly being performed to assist. Moreover, an actual autonomous reasoning agent in a complex domain will likely use far more extensive reasoning than the toy example given above, and there may be genuine computational issues associated with articulating and permitting argumentation over complete chains of deduction.

These problems are resolved in this thesis by providing a means to limit the initial complexity of arguments and then incrementally expand upon them as deemed necessary by agents. For example, in this case, Alanna may initially simply state her claim:

Alanna: I can be trusted with the library.

Benjamin can then explore Alanna's claim by requesting that Alanna elaborate upon it more, or by anticipating her reasoning, elaborating upon it himself and attacking that elaboration (at which point Alanna can counter-attack or describe how her reasoning is different from that presumed by Benjamin):

Alanna: I can be trusted with the library because I am a researcher in the field and have no reason to abuse your trust.

Benjamin: You cannot be a researcher, because you do not appear to have a research topic.

By expanding the space of argumentation only incrementally from a minimal starting space, unnecessary detail is dispensed with, and uncontroversial claims can be dealt with with minimal consideration.

Moreover, in keeping with our focus on fully distributed interactions, it follows that our mechanism must be fully distributed and tolerant of the vagaries of an asynchronous system, its product easy to disseminate from peer to peer. This requires that adequate regard is given to the challenges of communicating arguments in an asynchronous setting with many different agents involved. Such regard is duly given.

The ultimate purpose of our portrayal mechanism then is two-fold. Firstly, we seek to make interactions between autonomous agents more robust by requiring agents to ensure that their decisions are admissible to their peers (if not necessarily preferred by them). By accumulating evidence and challenging assumptions, it is hypothesized that

agents become more likely to validly achieve their objectives; the more a particular group of agents engages in such discourse, the greater the degree of synchronisation between their beliefs, and the greater the basis for further collaborations — in question is merely how efficiently this can be done. From the perspective of interaction design, the use of portrayals make it unnecessary to write protocols which explicitly specify certain processes of debate and negotiation, allowing for more generic, lightweight protocols.

Secondly, we wish to correct inconsistencies between the beliefs of an agent and its peers. We blame such inconsistencies primarily on the requirement in complex, dynamic environments for agents to independently make assumptions in order to make concrete decisions, and it is evident that interaction between agents provides a good opportunity to discover and correct any mistakes made. However simply throwing the beliefs of agents together into one giant belief revision problem is not feasible in a system which is in continual flux and which will therefore likely have changed before a solution can be produced. Fortunately, we can use the interactions engaged in by agents as a kind of opportunistic prioritisation mechanism for truth maintenance, on the basis that the constraints on interaction identify where consistency between agents is most vital. The use of argumentation to disseminate and process propositions as part of our portrayal mechanism can be shown to implement this opportunistic prioritisation.

1.3 Thesis Outline

Chapter 1 of this thesis has provided a backdrop for the rest of the thesis, introducing the contribution itself, as well as the problems it exists to address. The next part of this thesis provides the theoretical framework underpinning our contribution:

Chapter 2 examines the notion of multi-agent interaction, and in particular the kind of distributed, protocol-driven dialogues which we seek to augment with our contribution, providing a brief review of pertinent literature.

Chapter 3 explores the use of argumentation as a generic framework for modelling an agent's internal reasoning processes.

Chapter 4 builds upon Chapter 3 by bridging internal argumentation with social argumentation between peers. This chapter also defines formally the distributed decision problem which our portrayal mechanism exists to solve.

The second part of this thesis concerns itself with the specification and implementation of our distributed interaction portrayals:

Chapter 5 specifies the portrayal mechanism, defining the notion of a portrayal instance and the operations which agents can invoke to manipulate one. In this chapter, we specify how agents construct arguments within the confines of a portrayal, as well as how agents determine whether a portrayal has been adequately reconciled with their own beliefs.

Chapter 6 provides an in-depth example of the portrayal mechanism in action; this serves to temper the theoretical properties specified in earlier chapters with a demonstration of the practical benefits of augmenting an interaction with a portrayal.

Finally, **Chapter 7** concludes the thesis, discussing the practicalities of the contribution as well as providing an overview of possible future work.

1.4 Prior Publication

Some of the work in this thesis was presented at the Autonomous Agents and Multiagent Systems (AAMAS) conference in May 2010 [Martin et al., 2010]:

Martin, Robertson, Rovatsos. (2010). Opportunistic belief reconciliation during distributed interacions. In *Proceedings of the 9th Interactional Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 433–440.

Chapter 2

Autonomous and Distributed Agent Interactions

The purpose of interaction between agents within a distributed system is to propagate knowledge and to assemble new behaviours by coordinating the actions of individual peers. If performed correctly, agents will naturally assume roles which befit their particular expertise, whether a given role demands that an agent acts as oracle, coordinator, instrument of change or anything else. Whilst interactions may ultimately be orchestrated for the benefit of outside forces, an autonomous agent will be motivated by its own goals regardless of their provenance, and so can be expected to act upon its own initiative, collaborating with its peers to construct new interactions as they are deemed necessary.

Consider however interaction within an *open* distributed system. We define an open system here as one which allows the contribution of components from different sources which do not necessarily adhere to any particular model or design. Whilst the composition of the system as a whole may be a product of design along with the medium for communication, the nature of individual agents within the system is such that their behaviours cannot be predicted by their peers except after rigorous observation. Conflicting ideals and standards can produce agents with notably different internal processes. Combining the idiosyncrasies of heterogeneous agency with both a complex, often inaccessible environment and a dynamic agent population makes the coordination of arbitrary groupings of agents towards arbitrary ends a challenging problem to address.

Given then a population of autonomous agents within an open distributed system, it becomes necessary to provide a common framework in which interaction can be

controlled and directed on the volition of those agents participating, without unduly constraining agent autonomy beyond that which is necessary to ensure one of a number of recognisable outcomes. One approach is to formalise the social institutions which underlie certain classes of interaction, defining the protocols to which those agents should adhere if they want to act within such institutions. Having specified these social models for interaction, it then becomes necessary to disseminate them in a form which can be executed by amenable peers.

This chapter concerns itself with interaction choreography within a distributed system of autonomous, knowledge-based agents. In §2.1 we consider the very nature of interaction and dialogue. In §2.2 we consider how interactions can be modelled by agents, and how these models can be used to specify interactions in a distributed system as well as merely predict them. This allows us in §2.3 to provide a specification for distributed interaction based on an interaction protocol. Finally, in §2.4, we consider how the distribution of interaction affects the ability of interacting agents to make coherent decisions. All this serves to provide a formal description of the kind of interactions which we are interested in, and to define the problem which we seek to solve.

2.1 Interaction and Dialogue

An interaction is *'reciprocal action; action or influence of persons or things on each other'*.[1] This is superficially uncontroversial, but is also rather vague, which unfortunately seems to describe the definitions of interaction habitually used in multi-agent systems research. For example [Weiss, 2000] defines an interaction as any activity occurring between agents or between agents and their environment — an extremely broad definition which whilst not necessarily incorrect, fails to impart any significant insight into the nature of reciprocal activity. Given that we are concerned here with interactive symbolic reasoning, it seems somehow inadequate that the notion of interaction itself lacks the precision necessary to be defined symbolically.

A *run* of a distributed system records the observation over time of events occurring during a given execution of the system by the processes within it [Fagin et al., 1995, Halpern and Moses, 1990]. Such a run can be used to construct a logical time-line of all internal and external events which have affected a given system of agents over the course of its existence. An interaction can merely be seen then as any particular set of

[1] Courtesy of the *Oxford English Dictionary* [Simpson, 2006].

events drawn from such a time-line which exhibit certain properties. We shall define these properties as so:

Causality — Every event is an action by an agent which influences or is influenced by at least one other event in the interaction precipitated by another agent (events involve more than one agent and are all causally linked). *An event influences another if the former necessarily occurs prior to the latter.*

Connectivity — The set of events defining an interaction cannot be divided into two disjoint subsets in such a way that one subset does not influence the other (the selected events describe a single interaction). *A set of events influences another if there exists an event in the latter set which is influenced by an event in the former set.*

Whilst perhaps not strictly essential insofar as there exist communicative processes which one might wish to model which do not fulfil this criterion, there is also another property which could be said to be required of 'reciprocal action':

Participation — Every agent involved in the interaction both wields influence and is itself influenced over the course of events (an agent which merely dictates events or just passively observes them is not considered a participant of the interaction). *An agent influences another if the latter performs an action which is influenced by an action of the former.*

Here an event is essentially anything that 'happens' between agents over some period of time. Events are holistic; an event can describe the combination of many smaller events, in which case an interaction itself is an event (and thus interactions must themselves be holistic). We mainly concern ourselves however with events which are observable actions perpetrated by agents towards other agents. Thus we can formally define our own notion of interaction specifically for this thesis:

Definition 2.1 *An* **interaction** I *is a set of events* $\{\varepsilon_1, \ldots, \varepsilon_n\}$ *(where* $n > 1$*) which have been partially-ordered by a relation* $\prec$ *such that:*

- *Each event* ε *is an action by a set of agents A observed by a set of agents O. Both sets must be minimal under set inclusion (i.e. there is no agent in A which was not involved in perpetrating* ε*, and likewise there is no agent in O which did not observe* ε*).*
- *If* $\varepsilon_i \prec \varepsilon_j$ *then event* ε_i *necessarily occurred prior to* ε_j*.*

- *For every action $\varepsilon_i \in \mathcal{I}$ by a set of agents A_i observed by a set of agents O_i, there exists another action $\varepsilon_j \in \mathcal{I}$ such that either:*
  - ε_j *is an action by agents A_j where $O_i \cap A_j \neq \emptyset$ and $\varepsilon_i \prec \varepsilon_j$* (causality 1).
  - ε_j *is an action observed by agents O_j where $O_j \cap A_i \neq \emptyset$ and $\varepsilon_j \prec \varepsilon_i$* (causality 2).
- *Interaction $\mathcal{I}$ cannot be divided into two disjoint sets E_i and E_j in such a way that there exist no two events $\varepsilon_i \in E_i$ and $\varepsilon_j \in E_j$ for which either $\varepsilon_i \prec \varepsilon_j$ or $\varepsilon_j \prec \varepsilon_i$* (connectivity).
- *If an agent a is involved in enacting an event $\varepsilon_i \in \mathcal{I}$ such that $a \in A$, the set of actors, then there exists another action $\varepsilon_j \in \mathcal{I}$ for which $a \in O$, the set of observers. Likewise, if an agent o observes an event $\varepsilon_k \in \mathcal{I}$ such that $o \in O$, the set of observers, then there exists another action $\varepsilon_l \in \mathcal{I}$ for which $o \in A$, the set of actors* (participation).

The criteria of *causality*, *connectivity* and *participation* all rely on an abstract notion of 'influence'. Evidently a key consideration then in modelling interaction is one of pedantry — if an intelligent agent can be considered a product of its history, then it could be claimed that *every* event which occurs to that agent exerts influence on any and all future actions, suggesting that interactions can be found *everywhere*. Now this might be a perfectly reasonable philosophical assertion, but practicality dictates that we model interactions from a slightly less all-encompassing viewpoint, only considering events which effect a change of state which is discernible at whatever level of abstraction we choose to model. To that end, when speaking of an interaction between agents, there is often a focus on a particular class of activity.

A *dialogue* is a particular form of interaction wherein every event is a speech act [Austin, 1962, Searle, 1969] uttered by an agent which then exerts some illocutionary force upon at least one listener. As such, a dialogue can be considered to be a collection of messages exchanged between peers (in multi-agent systems often drawn from a performative language like KQML [Finin et al., 1994] or FIPA-ACL [O'Brien and Nicol, 1998]), where the illocutionary force of messages can be determined from the influence they wield on interaction. We can easily adapt Definition 2.1 for dialogue:

Definition 2.2 *A* **dialogue** $\mathcal{D}$ *is a set of speech acts $\{m_1, \ldots, m_n\}$ (where $n > 1$) which have been partially-ordered by a relation $\prec$ such that:*

- *Each speech act m is a message from an agent s to a set of recipients R.*
- *If $m_i \prec m_j$, then message m_i necessarily is received by at least one agent in R prior to m_j* (asynchronicity).[2]
- *For every message $m_i \in \mathcal{D}$ from an agent s_i to agents R, there exists another message $m_j \in \mathcal{D}$ such that either:*
  - *m_j is sent by an agent $s_2 \in R$, and $m_i \prec m_j$* (causality 1).
  - *m_j is received by agent s_i, and $m_j \prec m_i$* (causality 2).
- *Dialogue $\mathcal{D}$ cannot be divided into two disjoint sets E_i and E_j in such a way that there exist no two separate events $m_i \in E_1$ and $m_j \in E_2$ for which either $m_i \prec m_j$ or $m_j \prec m_i$* (connectivity).
- *If an agent s sends a message $m_1 \in \mathcal{D}$, then there exists another message $m_j \in \mathcal{D}$ for which $s \in R$, the set of recipients. Likewise, if an agent $r \in R$ receives a message $m_k \in \mathcal{D}$, then there exists another message $m_l \in \mathcal{D}$ which has been sent by r* (participation).

As a form of interaction, dialogue is of particular interest because it is purely and directly communicative; as such it is the focus of most research into interaction (from [Searle, 1969] to [Walton and Krabbe, 1995] to [Robertson et al., 2008] for example). By focusing on dialogue, we no longer need concern ourselves about indirectly influencing agents by the careful stacking of pebbles or the laying of pheromone trails, and instead consider only the near-immediate transfer of information between two peers. Additionally, dialogue has a non-recursive base unit — the speech act — which means we do not have to worry about the holistic nature of abstract events, and can simply measure the evolution of an agent system's state message by message.

Example 2.1 *Consider a scenario in which an agent Alanna desires data which can be found only in a specific library, one which is only accessible to a privileged few. Alanna knows of another agent Benjamin, who is a patron of that library. Alanna decides therefore to enlist Benjamin's support to acquire the access privileges she needs. This leads to the following (very simple) dialogue:*

1. *Alanna requests Benjamin's assistance; "Benjamin, could you help me get access to the library?"*

[2] This definition accounts for asynchronous communication by allowing response by one recipient of a message prior to the reception of the message by all intended recipients.

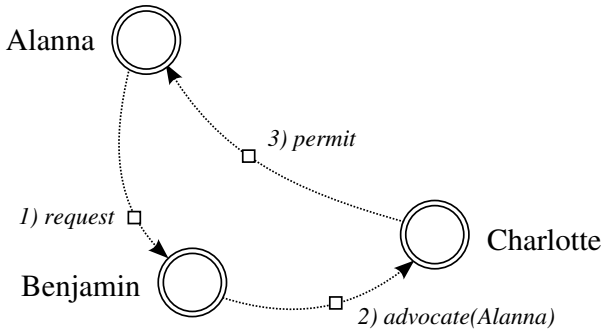


Figure 2.1: *A dialogue between agents Alanna, Benjamin and Charlotte, as described in example 2.1.*

2. *Benjamin recommends Alanna to Charlotte, the controller of the library; "Charlotte, I recommend that Alanna be given access to the library."*

3. *Charlotte then grants Alanna permission; "Alanna, you may access the library."*

Consequent to Alanna's initial request, each illocution is predicated upon the previous one (demonstrating causality and connectivity). Moreover, it is clear that each agent both acts and is acted towards (demonstrating participation).

Whilst many of the statements made over the course of this chapter can be seen to apply equally to any form of interaction, ultimately we are concerned in this thesis with interaction dialogues — acts of agent coordination driven by the exchange of messages. Thus, any statement made about interaction in the following sections should, by default, be considered to apply to dialogue-type interactions first and foremost.

It is not enough just to able to identify the presence of interaction however — we want to be able to model it as well. If an agent can model different types of potential interaction in order to identify the factors which differentiate one interaction from another, then that agent can plan for an interaction based on those factors which apply to its current circumstances.

2.2 Interaction Models and Protocols

An agent models possible interactions based on the expected responses of its peers to particular events occurring under different environmental conditions. Given an interaction model, an agent can use that model to predict the course of an interaction and

so choose to act in such a manner as to ensure a particular desired outcome. In a distributed system however, it is often impossible to predict with certainty the responses of peers, partially because of uncertainty about the true state of the environment and partially because the nature of the peers themselves is not always fully understood and thus their actions become intrinsically unquantifiable. Taken to a logical extreme, it might seem necessary for agents to produce arbitrarily complex contingency plans in order to cover all possibilities. This approach is clearly intractable.

Interaction can be made more predictable however by the use of interaction protocols.[3] In this context, an interaction protocol is basically an interaction model which has been in some way published in some standard format, such that it can then be used to *specify* interaction rather than merely predict it. If agreement can be obtained to the effect that all agents will only act in adherence to a given shared protocol for some particular interaction, then any agent involved in that interaction can more accurately predict the responses of its peers and the outcome which is likely to unfold because of them. All that would remain then is the problem of accurately determining the system state so as to adequately inform the decisions made by agents during interaction.

2.2.1 Modelling Interaction

If the initial state of a multi-agent system (which we will take here to include agent programming) is known, then it is possible given a run of the system to determine the state of that system at all points in that run, based on the cumulative state changes brought about by observation of events. Evidently, it must also be possible to determine how a specific interaction influences the system state as well. It is a mistake however to casually assume that any given interaction occurs in isolation — we can see this by considering the ease with which we can identify multiple overlapping interactions in any sufficiently involved system of agents. Even considering solely interaction dialogues, there will be any number of events external to the interaction which will influence agents and the course of dialogue without being formally part of it.

As such, whilst the system state can change because of the events which compose an interaction, it can also change in spite of them. There will also be events not part of an interaction which technically change the system state, but which have no actual influence on the outcome of an interaction whatsoever. We can mitigate the obfuscating

[3] In this chapter we use the term ‘protocol’ quite broadly. In [Maudet and Chaib-draa, 2002, Flores and Kremer, 2002] for example, a distinction is made between *dialogue protocols* and *conversation policies*, a distinction we do not make here.

effect of irrelevant events however by differentiating the interaction state from the system state. We can also abstract aside irrelevant elements of the concrete system state in favour of just the portion of the state which is pertinent to the interaction at hand. By doing these things, we can more clearly see the course of interaction and identify those events which actually affect it.

An *interaction model* is, in essence, simply a specification of some set of valid transitions between abstract states. Given a sufficient description of a system state, it is possible to determine the interaction that would then occur. An interaction model can be used to predict the outcome of ongoing interactions, but can equally be used to analyse them after they occur or to guide an agent through a future interaction:

Definition 2.3 *An* **interaction model** $\mathcal{M}$ *defines a set of transition relations* $(\mathbb{S}_i, E, \mathbb{S}_f)$ *where:*

- $\mathbb{S}_i$ *abstractly describes some initial system state.*
- $\mathbb{S}_f$ *abstractly describes some final system state.*
- E *is a set of partially-ordered events which would allow the transition of a system from state* $\mathbb{S}_i$ *to state* $\mathbb{S}_f$*.*

An interaction $\mathcal{I}$ *is* **modelled** *by an interaction model* $\mathcal{M}$ *if there exists a series of transitions* $(\mathbb{S}_0, E_1, \mathbb{S}_1), (\mathbb{S}_1, E_2, \mathbb{S}_2), \ldots, (\mathbb{S}_{n-1}, E_n, \mathbb{S}_n)$ *(where* $n > 0$*) such that* $\mathcal{I} \subseteq \bigcup_{j=1}^{n} E_j$ *(where* $E_j \prec E_{j+1}$*) and for every transition relation* $(\mathbb{S}_{j-1}, E_j, \mathbb{S}_j)$*, there exists a corresponding transition relation* $(\mathbb{S}_i, E_j, \mathbb{S}_f) \in \mathcal{M}$ *wherein* $\mathbb{S}_i \sim \mathbb{S}_{j-1}$ *and* $\mathbb{S}_f \sim \mathbb{S}_j$ *as defined in Definition 2.4 below.*

Transition relations specify possible outcomes of certain sequences of events occurring given a particular system state. We do not ascribe any particular format to a system state description — for now, it is sufficient to think of such descriptions as restrictions on the possible worlds in which a given state transition can occur. We will also limit our consideration to *global* states, leaving aside the notion of local state until later:

Definition 2.4 *Two system state descriptions* $\mathbb{S}_1$ *and* $\mathbb{S}_2$ *are* **compatible** *if the intersection of possible worlds described by* $\mathbb{S}_1$ *and* $\mathbb{S}_2$ *is not empty (i.e. it is possible for both state descriptions to describe the same world). This fact is denoted* $\mathbb{S}_1 \sim \mathbb{S}_2$*.*

Note that there may be many ways to construct the same interaction from a set of transition relations. In practice, any interaction model would likely be specified using some

process calculus from which transition relations can be derived from an enumeration of the different cases or outcomes accounted for (e.g. [Robertson, 2004, Walton, 2004a]).

Example 2.2 *Consider how Alanna might model the dialogue of Example 2.1. Informally:*

> *I need access to the library. I can request the assistance of an existing patron of the library. If my chosen patron trusts me, then he will recommend me to the library controller, otherwise he will decline to assist me, ending our dialogue. Upon receiving a recommendation, the controller will grant me permission if she considers me to be eligible, at which point I have achieved my goal. Otherwise, the controller will refuse me, again ending dialogue.*

This informal model can be decomposed into a number of state transitions of the form $(\mathbb{S}_i, E, \mathbb{S}_f)$. *For example:*

$\mathbb{S}_i$ = *The library is inaccessible, but Benjamin is a patron of that library.*

E = {*Alanna: "Benjamin, could you help me get access to the library?"*}

$\mathbb{S}_f$ = *The library is inaccessible, Benjamin is a patron of that library, and Alanna has requested that Benjamin recommend that she be given access to it.*

Another example:

$\mathbb{S}_i$ = *Benjamin has recommended Alanna to the controller of the library, and Alanna is eligible for access to that library.*

E = {*Controller: "Alanna, you may access the library."*}

$\mathbb{S}_f$ = *The controller has permitted Alanna to access the library, and so the library is accessible.*

Using these state transitions, Alanna can model the dialogue of Example 2.1 using any series of transitions wherein each final state is compatible with the initial state of the next transition, and where the events generated by such a series of transitions subsume the dialogue.

As already stated the outcome of an interaction is dependent on the system state. The purpose of an *interaction state* is to describe an ongoing interaction and to relate that interaction to some underlying interaction model such that it can be used to determine possible system states:

Definition 2.5 *An* **interaction state** $\mathcal{S}$ *for an interaction modelled by some interaction model* $\mathcal{M}$ *can be described by a tuple* $(H, \Delta, \mathcal{M})$ *where:*

- *H is the interaction as transpired so far; H is a set of events, partially-ordered by a relation $\prec$ as per Definition 2.1.*

- *Δ describes the obligations outstanding on interaction in order to obtain certain tangible outcomes; Δ can be viewed as a set of possible event sets, each partially-ordered like H, but each describing an exclusive outcome.*

- *For every event set $F \in \Delta$, it is the case that $(H \cup F)$ is an interaction as per Definition 2.1 with the caveat that there is no event $\varepsilon_f \in F$ and event $\varepsilon_h \in H$ such that $\varepsilon_f \prec \varepsilon_h$ (i.e. F may be caused by H, but H is never caused by F).*

- *For every set $F \in \Delta$, the interaction $(H \cup F)$ is modelled by $\mathcal{M}$ as per Definition 2.3.*

An interaction state serves then to describe what has already occurred and what may yet occur given that the completed interaction is expected to be one defined by the selected interaction model. Uncertainty as to the final interaction exists because the system state may not be known in its entirety, and external events may change the system such that interaction cannot proceed according to the given model — instead, the outcome of the interaction will be determined by the transitions available in the interaction model given the system state at the point of execution.

In order to determine whether a modelled interaction is ongoing within a given system, we need to be able to determine whether the interaction state models the system state, in analogous fashion to how an interaction model might model a specific interaction. To be able to do this for non-trivial cases, it must be possible to access part of the run of the system prior to the specified system state in order to identify any portion of the interaction which has already occurred during that run:

Definition 2.6 *A* **transition history** *$h(\mathbb{S})$ of some system state $\mathbb{S}$ is a (partial) description of the transitions made by a system into state $\mathbb{S}$, where:*

- *$h(\mathbb{S})$ can be considered to be a transition relation $(\mathbb{S}_0, H, \mathbb{S})$, where H is a partially-ordered set of the events which have happened since some initial state $\mathbb{S}_0$ allowing the transition to state $\mathbb{S}$.*

- *A transition $(\mathbb{S}_i, E, \mathbb{S}_f) \in h(\mathbb{S})$ if and only if there is a sub-transition $(\mathbb{S}'_i, E', \mathbb{S}'_f)$ of $(\mathbb{S}_0, H, \mathbb{S})$ (as per Definition 2.3) such that $\mathbb{S}_i \sim \mathbb{S}'_i$, $\mathbb{S}_f \sim \mathbb{S}'_f$ and $E \subseteq E'$.*

Once we are able to confirm that the past events in an active interaction have occurred in a system, it then remains to show that the system is in a state which allows the interaction to be completed.

Definition 2.7 *An interaction state $\mathcal{S} = (H, \Delta, \mathcal{M})$* **models** *some system state $\mathbb{S}$ if and only if for every $F \in \Delta$ it is the case that $(H \cup F)$ is modelled by $\mathcal{M}$ using a sub-transition $(\mathbb{S}_i, H, \mathbb{S}) \in h(\mathbb{S})$ such that $\mathcal{S}$ describes an ongoing interaction in $\mathbb{S}$.*

Let us return then to our running example:

Example 2.3 *Assume that Alanna initiates the dialogue described by Example 2.1. Assume that the initial system state $\mathbb{S}_0$ can be described as follows:*

> *The library is inaccessible to Alanna, but Benjamin is a patron of that library, and Benjamin trusts Alanna.*

The initial interaction state $\mathcal{S}_0 = (\emptyset, \Delta_0, \mathcal{M})$, where $\mathcal{M}$ is the interaction model described in Example 2.2 and Δ_0 describes two possible dialogues:

1. *Alanna requests help from Benjamin, Benjamin recommends Alanna to the library controller, and the controller grants Alanna permission.*
2. *Alanna requests help from Benjamin, Benjamin recommends Alanna to the library controller, but the controller refuses to grant Alanna permission.*

$\mathcal{S}_0$ models $\mathbb{S}_0$ because for all interactions described by Δ_0, it is possible for them to occur given state $\mathbb{S}_0$; Alanna does not expect the scenario in which Benjamin declines to assist Alanna, because she knows that Benjamin trusts her, and (according to her model) will therefore recommend her to whoever the library controller is.

According to $\mathcal{M}$ (and Example 2.2), Alanna is able to request help from Benjamin, transitioning into a new state $\mathbb{S}_1$. This moves the event 'Alanna requests help from Benjamin' to the history H_1 of interaction state $\mathcal{S}_1$, and leaves Δ_1 describing the possible remaining events:

1. *Benjamin recommends Alanna to the library controller, and the controller grants Alanna permission.*
2. *Benjamin recommends Alanna to the library controller, but the controller refuses to grant Alanna permission.*

$\mathcal{S}_1$ models $\mathbb{S}_1$ because $(\mathbb{S}_0, H, \mathbb{S}_1) \in h(\mathbb{S}_1)$ and all interactions $(H_1 \cup F)$ for $F \in \Delta_1$ are possible. Suppose that Benjamin then recommends Alanna to Charlotte. The system state would then be one in which Benjamin had just made his recommendation, and

Charlotte would then grant or reject Alanna's request. It cannot be predicted which would happen however, because Alanna does not even know if she is eligible, and so either outcome would result in an interaction state modelled by the updated system state.

Unmodelled events can lead to an interaction state no longer modelling the system state, such that interaction cannot proceed without violating in some way the interaction model (i.e. there are no valid transitions from the current state).

Example 2.4 *The interaction model given in Example 2.2 only accounts for a very limited set of eventualities. Consider a few of the unmodelled events which might occur after Alanna requests help from Benjamin:*

- *Benjamin might decline to assist Alanna,* despite *trusting her, because such an act might interfere with some prior obligation.*
- *Instead of contacting Charlotte as expected, Benjamin might render assistance by some other means, such as offering to acquire the desired information for her.*
- *Library controller Charlotte might request additional information from Alanna prior to determining eligibility — the interaction as modelled will not proceed until Alanna responds to Charlotte's query.*
- *Benjamin or Charlotte responds to Alanna's request in a manner which Alanna simply does not understand.*

Any of these events will require Alanna to reconsider her interaction model if she is to respond intelligibly.

There are four basic responses an agent can take upon discovering that the state of an interaction does not model the system state; the interaction can be aborted, the agent can wait until the system returns to a state compatible with the interaction, the interaction can be 'repaired' or the interaction model can be revised.

Aborting an interaction is best kept as a last resort. Not only does the agent not acquire any of its goals, but this response may violate the agent's obligations (at least as they are viewed by its peers). The consequences of this can vary from being mostly harmless to being severely damaging to an agent reputation, resulting in loss of privileges or greater difficulty in obtaining the cooperation of other peers in future.

In a dynamic system, the state of the world changes, and sometimes it changes back. A group of agents could be involved in an interaction which can only proceed in a

given state, temporarily stopping activity whenever that state does not hold. Obviously however, there are scenarios where the system will *not* return to a state compatible with an interaction, at least not without the intervention of the agents involved.

'Repairing' an interaction entails actively shifting the system state into one which the given interaction model can provide a valid transition relation for. The idea is that agents in an interaction can engage in some temporary digression in order to restore the system state to one in which interaction can continue. For example, an agent can perform an action which changes its environment to a more desirable state, or it could stop to gather or supply additional information for its peers which might allow the interaction to continue (for example, in the case where Charlotte makes an unexpected request of Alanna, Alanna might be able to repair the interaction simply by fulfilling the request). Note that any auxillary interactions may themselves need to be modelled, albeit perhaps independently of the main interaction.

Sometimes however, the unexpected event is not simply some distraction which can be cleanly dealt with prior to continuing interaction. Sometimes the unexpected event indicates the unfolding of an entirely unplanned outcome (for example, Charlotte might offer Alanna an alternative means to acquire the information which motivates her actions). Revising an interaction model is basically a matter of adding new transition relations to the model such that the unfolding interaction is adequately specified. The fundamental difficulty is that an agent may not be in a position to comprehend the consequences of a hitherto unexpected event on the system state, in which case it will not be able to extrapolate the remainder of the interaction, nor determine the outcome. Essentially, the agent would need an understanding of the interaction that goes beyond the model given — by, for instance, being able to classify speech acts by illocutionary force (see the discussion of agent communication languages such as KQML [Finin et al., 1994] below). We generally assume otherwise, since for our purposes this simply implies that there exists another, greater interaction model available to an agent which happens to subsume the one used.

Of the four possible responses just described, one is essentially surrender, another involves doing nothing and yet another requires either considerable creative intelligence on the part of the interacting agent, or considerable over-engineering of interaction models commensurate with the complexity of the system in which interaction occurs. Only interaction repair seems generically practical. Ideally though, we would simply prefer that our interaction models accurately described system behaviour in the first place, such that interaction states model system states more often than not. In the

next section then, we compound on our woes by pointing out how difficult this really is in practice.

2.2.2 Institutionalising Distributed Interaction

In a distributed multi-agent system, agents are autonomous processes, with their own state and programming. Any given agent in the system is unlikely to be able to access the local states of their peers, nor are they necessarily able to perceive all parts of the environment in which the distributed system is situated. On the other hand, the nature of interaction is such that agents are only expected to respond to events which they can directly observe. Dialogue in particular can be seen as a series of messages exchanged from one peer to another — with any response dependent on only on the message and an agent's beliefs at the time of reception. Thus it should be possible to create models of interaction wherein all transitions of system state can be localised to particular agents; these local transitions can then be used to reconstruct a description of a complete interaction.

We consider a *distributed* interaction model as being composed then of local agent transition relations, where each transition relation describes the events which must be observed in order for a particular (abstract) agent to transition from one local state to another, generating further events as a consequence:

Definition 2.8 *A* **distributed interaction model** $\mathcal{M}$ *defines a set of local transition relations* $(\sigma, \mathbb{L}_i, E_o, E_g, \mathbb{L}_f)$ *where:*

- $\sigma \in \Sigma$*, where* Σ *is the set of identifiers for agents acting in particular roles in* $\mathcal{M}$ *.*
- $\mathbb{L}_i$ *abstractly describes the system state local to* σ *prior to generating* E_g*.*
- $\mathbb{L}_f$ *abstractly describes the system state local to* σ *after observing* E_o*.*
- E_o *is the set of events observed by* σ *which allow the transition of the local state of* σ *from* $\mathbb{L}_i$ *to* $\mathbb{L}_f$*.*
- E_g *is the set of events generated by* σ *over the course of the transition of the local state of* σ *from* $\mathbb{L}_i$ *to* $\mathbb{L}_f$*.*
- $\varepsilon_o \prec \varepsilon_g$ *for all events* $\varepsilon_o \in E_o$ *and* $\varepsilon_g \in E_g$*, where* $\prec$ *is a temporal ordering as per Definition 2.1.*

$\mathcal{M}$ also defines a number of environment transition relations $(\mathcal{E}, \mathbb{L}_i, E_o, E_g, \mathbb{L}_f)$, which are defined as above except that $\mathcal{E}$ represents the system beyond Σ as a single entity.

For system state transitions, events are fully internalised within the system, but because local transition relations focus on individual agents, there is a need to distinguish between events observed and events generated by those agents. It is also necessary to note those events observed or generated outwith those agents — the 'environment' in this case includes anything not immediately connected to the interaction being modelled, including agents which do not have roles in that interaction. This is important because local transitions are focused on the private states of individual agents, and events directed towards the environment are required for interactions to exact change to the system state beyond those private states:

Definition 2.9 *A system state transition $(\mathbb{S}_i, E, \mathbb{S}_f)$ is* **described by a distributed interaction model** *$\mathcal{M}$ if and only if either:*

- *There exists a set Γ of local transitions $(\sigma, \mathbb{L}_i, E_o, E_g, \mathbb{L}_f)$ and environment transitions $(\mathcal{E}, \mathbb{L}_i, E'_o, E'_g, \mathcal{S}_f)$ such that:*
  - *E is the union of all observed events E_o in Γ minus all generated events E'_g in Γ.*
  - *E is also the union of all generated events E_g in Γ minus all observed events E'_o in Γ.*
  - *System state $\mathbb{S}_i$ is the combination of all local states $\mathbb{L}_i$ in Γ.*
  - *System state $\mathbb{S}_f$ is the combination of all local states $\mathbb{L}_f$ in Γ.*
- *$(\mathbb{S}_i, E, \mathbb{S}_f)$ can be constructed from a series of sub-transitions $(\mathbb{S}_i, E_1, \mathbb{S}_1), \ldots, (\mathbb{S}_{n-1}, E_n, \mathbb{S}_f)$ (where $n > 1$) and every such sub-transition is described by $\mathcal{M}$ as per this definition.*

An interaction $\mathcal{I}$ is modelled by $\mathcal{M}$ if there exists a transition $(\mathbb{S}_0, E, \mathbb{S}_n)$ described by $\mathcal{M}$ such that $\mathcal{I} \subseteq E$.

Thus, provided that we can distribute (and then reintegrate) the interaction state, we can use the definitions of the previous section to describe distributed interaction.

It is not simply the system state which is difficult to access in a distributed system. It is also more difficult to quantify the set of possible events which might provoke a transition, because individual agents are not necessarily cognisant of the full

capabilities of their peers, or even the full underlying dynamics of the broader environment in which a distributed system might exist. For interaction dialogues, this problem can be resolved by the use of agent communication languages such as the Knowledge Query Management Language (KQML) [Finin et al., 1994] and the Foundations of Intelligent Physical Agents Agent Communication Language (FIPA ACL) [O'Brien and Nicol, 1998].

Agent communication languages formalise dialogue by classifying the permissible types of illocution an agent can make and defining semantics for each type, essentially quantifying the events which can be generated as a part of an interaction dialogue. KQML defines a strict set of *performatives* (such as tell, ask-if and reply) which agents can use to communicate with one another. Every message generated in KQML must be of a given performative type and provide a certain set of parameters, including references to the logical language and ontology used.

Example 2.5 *If the dialogue described in earlier examples in this chapter was conducted using an agent communication language like KQML, messages would arrive looking something like this:*

```
(ask-if
        :sender     Alanna
        :receiver   Benjamin
        :language   Prolog
        :ontology   Resource-Patronage
        :content    "recommend(access(Alanna, library))"
        ...)
```

Where in this case, Alanna is asking Benjamin if he could recommend that Alanna be granted access to the library. This expressed using a Prolog proposition and a hypothetical ontology.[4]

FIPA ACL is the result of a consortium effort to create a standard agent communication language, based on examination of KQML and an interface language known as Arcol [Breiter and Sadek, 1996] developed for France Telecom. It retains many of the principles of KQML, but culls many 'unnecessary' performatives and grants greater capacity for combining performatives. There exist many variants of KQML and FIPA ACL [Labrou et al., 1999], and the two languages can be seen as representative of that approach to agent communication as a whole.

[4]There does exist a Knowledge Interchange Format (KIF) [Genesereth and Fikes, 1992], which is specifically designed for information distribution, and which is often used with KQML, both being developed as part of the Knowledge Sharing Effort (KSE) [Patil et al., 1992].

One criticism of the performative language approach however is that it relies on *mentalistic* rather than *social* reasoning [Singh, 1998, Wooldridge, 2000] — that is, agents still need to reason about the internal beliefs of other agents with no guide as to whether they comprehend events in the same way as their peers, rather than relying on social norms as a means to bind peers to a shared social contract. In terms of our formalism, agents might be able to quantify the 'events' of dialogue, but they will still have to model transitions for every possible response to a given performative an arbitrary peer *might* make over the course of interaction depending on their local state (which they cannot ever be certain of).

There is another approach available however. Consider this. Agents can model interactions in terms of transitions between interaction states. By analysis of these transitions, an agent can determine the circumstances under which a given outcome will transpire. An agent can then apply an interaction model in order to ensure desired outcomes in future interactions, where those interactions adhere to the model. In a distributed system however, an agent does not have direct access to the programming of its peers, and thus may not be able to accurately predict the actions which its peers might take during an interaction, making it unlikely that it will be able to produce interaction models for arbitrary interactions on demand.

For most practical interactions there are not usually many qualitively different functional outcomes however. Out of those outcomes which do exist, many are simply graceful failures, where one party in an interaction declines to proceed further due to adverse circumstances or personal disinterest. The difficulty in modelling arbitrary interactions is not really a result of too many possible outcomes, but from the difficulty inherent in recognising that a particular arbitrary event indicates a move towards a particular functional outcome.

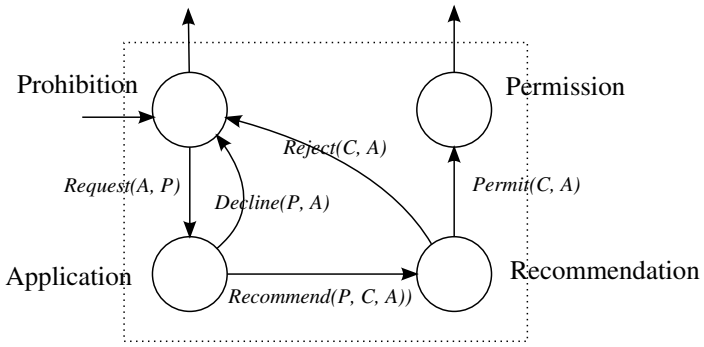
It would be advantageous if the interaction models used by agents to *predict* interaction could be published in some common format and then used to *specify* interaction instead — this is the idea behind agent protocol languages. An interaction protocol formalises a particular set of interactions by specifying the 'correct' responses to particular events, limiting the options available to agents to only those which allow an interaction to converge upon one of a finite set of monitorable outcomes; each outcome comes with conditions and obligations imposed upon it which are made known to all agents bound by the protocol. Thus, if an agreement can be extracted from all agents participating in a given interaction to the effect that they will all adhere to a given protocol, then that interaction becomes easier to model insofar as the interaction

becomes explicitly the product, rather than the source, of the model. This approach to interaction is less dependent on the internal states of agents, being more concerned with social commitment [Jennings, 1993].

There has been a great deal of research into particular types of agent interaction protocol, such as contract nets [Davis and Smith, 1983], blackboard systems and general negotiation dialogues [Rosenschein and Zlotkin, 1994, Rahwan et al., 2003] — a survey of general types can be found in [Huhns and Stephens, 2000]. All these systems share the same inherent (dis)advantage — they strictly limit the events and state transitions being modelled, and compel agents to only generate those events and transitions. Thus, whilst providing a practical means to conduct distributed interaction, they do not permit circumstances not foreseen by the protocol designer. Nonetheless, the use of protocols is deemed the best basis for agent interaction without a massive increase in the creative intelligence of agents.

Of interest to the multi-agent community then is the development of languages and tools to specify and publish different kinds of interaction protocol. One such development is electronic institutions [Esteva et al., 2001]. Institutions define collections of scenes, which each define a protocol by which agents should conduct themselves within the given scene, as well as the norms under which such a protocol should be conducted [García-Camino et al., 2005]. A number of tools have been developed to aid in the specification of such institutions [Esteva et al., 2002, Esteva et al., 2004].

Example 2.6 *Our running example can be seen as a scene within an electronic institution describing the use of a library as a social utility. The scene could be described using finite state automata, showing the states between which agents, acting in their given role, would transition in response to certain locutions:*



The scene begins in the 'prohibited' state, but can move into the 'application' state if Alanna can send a request message to a possible patron. That patron can either decline the request, returning the scene to the application state, or he can make a

recommendation to the library controller, moving the scene into the 'recommendation' state. The scene can be left either from the prohibition state (if Alanna is unable to achieve her objective) or from the 'permission' state (if Alanna gets the access she needs).

A flaw inherent in [Esteva et al., 2001], as specified, is a lack of proper distribution of the protocols defined. Institutions are policed by overseers, which determine whether or not a given agent is acting in accordance with its designated role. One would prefer that protocols could be enacted without such oversight, in a fully decentralised fashion.

The key to distributing an interaction protocol, and thus distributing an interaction model amongst a group of peers, is the ability to disseminate to each agent which wishes to adhere to the protocol the information necessary to allow them to perform their role in the interaction without requiring access to the entire system. At the same time, by disseminating each agent's role specification to *all* agents in an interaction, even if other peers cannot execute that role, all agents become aware of the social consequences of certain events observed during interaction [Osman, 2007], and can tell if their peers are acting in accordance with their commitments. We justify the removal of the requirement that agents be policed by observing that a fair interaction protocol ensures that it is in the interests of agents to follow the protocol. Moreover, common knowledge of a protocol ensures that agents which violate their commitments will be recognised, and this can be factored into future match-making of agents to particular roles in interaction [Lambert and Robertson, 2005].

The Lightweight Coordination Calculus [Robertson, 2004] and its sister calculus Multi-Agent Protocols [Walton, 2004a] use this approach. The strength of LCC (and MAP) lies in its ability to choreograph a group of agents at a high level of abstraction, with arbitrarily complex requirements hidden behind simple declarative constraints. We have already mentioned LCC in Chapter 1, and shown an example of an LCC protocol in Example 1.1. LCC makes no commitment as to how protocols are distributed, or how agents can be found to fulfil certain roles however — however there have been numerous extensions made to LCC which demonstrate how it can be used in a number of practical domains [Robertson et al., 2008]. In particular, the recent *OpenKnowledge* project [Siebes et al., 2007] has provided an entire framework for multi-agent service composition and execution which uses LCC to coordinate interactions.

2.3 A Formal Specification of Distributed Interaction

Having reviewed a few of the available approaches to interaction, we now concentrate on distributed dialogues specified by a common interaction protocol, as was initially described in Chapter 1. We provide here an abstract logical specification of how such dialogue can be conducted in a distributed system — we do this so that later in Chapter 5, we can demonstrate how our portrayal mechanism might operate alongside an interaction by showing how it interfaces with the specification defined here. We shall also show how our specification can be applied to the Lightweight Coordination Calculus [Robertson, 2004] — being as it is exemplary of the approach to interaction we advocate in this thesis, and pivotal to optimal understanding of the examples of interaction found in Chapter 6.[5]

As in §2.2.1, we consider interaction, at its most abstract, as a set of events motivating the transition of an agent system to a new state:

Definition 2.10 *An interaction ι brings about a system state $\mathbb{S}_f$ if the execution of ι causes a complete transition of an interaction state $\mathcal{S}_i$ modelling some prior system state $\mathbb{S}_i$ into a new interaction state $\mathcal{S}_f$ which models $\mathbb{S}_f$:*

$$\mathsf{interaction}(\mathbb{S}_i, \iota, \mathbb{S}_f) \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{models}(\mathbb{S}_i, \mathcal{S}_i) & \wedge \\ \mathsf{transition}(\mathcal{S}_i, \iota, \mathcal{S}_f) & \wedge \\ \mathsf{models}(\mathbb{S}_f, \mathcal{S}_f) & \wedge \\ \mathsf{closed}(\mathcal{S}_f) & \end{array} \right)$$

Where:

- $\mathsf{models}(\mathbb{S}, \mathcal{S})$ *is true if interaction state $\mathcal{S}$ models system state $\mathbb{S}$ as per Definition 2.7.*
- $\mathsf{transition}(\mathcal{S}_i, E, \mathcal{S}_f)$ *is true if the set of events E causes a transition from interaction state $\mathcal{S}_i$ into interaction state $\mathcal{S}_f$.*
- $\mathsf{closed}(\mathcal{S})$ *is true if $\mathcal{S}$ describes a complete interaction such that $\mathcal{S} = (\iota, \emptyset, \mathcal{M})$ as per Definition 2.5, where interaction model $\mathcal{M}$ is specified by some protocol.*

Given an initial system state $\mathbb{S}_i$, we state that an agent can bring about interaction ι if it can select a protocol compatible with $\mathbb{S}_i$ (such that initial interaction state $\mathcal{S}_i$ models

[5] The formal model specified in this section is adapted from that of [Robertson et al., 2008], but differs in that it specifies a parallelised rather than linear model for multi-agent coordination, which we believe to be more generically useful.

$\mathbb{S}_i$) and perform a complete enactment of that protocol (generating $\mathcal{I}$ as we transition from $\mathcal{S}_i$ to a closed interaction state $\mathcal{S}_f$) bringing about a new system state $\mathbb{S}_f$ (being a state modelled by $\mathcal{S}_f$).

This is a declarative depiction of an interaction; it can be used to identify that an interaction described by a given interaction model has occurred within a system by identifying a series of system state transitions which match one described by the model, or it can be taken as an abstraction of the process by which an interaction model can guide the transition of a system to a new state, with an interaction as its by-product — our concern here is primarily with the latter interpretation.

Let us now examine more closely the different parts of our specification.

2.3.1 Initiating Interaction

The first requirement of protocol-based interaction is that there exists an applicable protocol available to a motivated agent prior to interaction (in practical terms, this requirement is best fulfilled by having either a publishing service for protocols, or a means to synthesise protocols). Agents can then determine which of the available protocols is actually applicable to its current circumstances, and which of those protocols best serves its goals.

Definition 2.11 *An interaction state $\mathcal{S}$ models a system state $\mathbb{S}$ if $\mathcal{S}$ is the initial state of an interaction based on a protocol $\mathbb{P}$ selected by an agent σ which it considers to be applicable to $\mathbb{S}$:*

$$\mathsf{models}(\mathbb{S},\mathcal{S}) \quad \leftarrow \quad \mathsf{selection}(\sigma,\mathbb{S},\mathbb{P}) \wedge \mathsf{initial\_state}(\mathbb{P},\mathcal{S})$$

Where:

- $\mathsf{selection}(\sigma,\mathbb{S},\mathbb{P})$ *is true if an agent σ would select the protocol $\mathbb{P}$ for a new interaction given the system state $\mathbb{S}$.*
- $\mathsf{initial\_state}(\mathbb{P},\mathcal{S})$ *is true if $\mathcal{S} = (\emptyset,\Delta,\mathcal{M})$ according to Definition 2.5 such that $\mathcal{M}$ is the interaction model specified by $\mathbb{P}$ and Δ describes all possible outcomes of executing an interaction according to $\mathbb{P}$.*

The selection of a protocol is based on what an agent knows of the system state, biased by its goals. We do not concern ourselves with exactly how this is done, accepting that it is part of the internal programming of the agent. We do need to concern ourselves

however with how an agent can select an interaction state which will model the system state, despite being (presumably) unable to access that entire state.

Fundamentally, a protocol should not make too many assumptions about the system from the start, so that it can gracefully bring interaction to a close if circumstances are different from those assumed by the initiating agent (this can be as simple as providing a way to close interaction immediately should an initial constraint fail). A protocol should also be fair to all peers involved; opportunity must be given to agents whenever applicable to decline committing to some course of action which they believe to be incompatible with their current beliefs or commitments. The motivation for the agent initiating an interaction to select a fair protocol is simply one of enlightened self-interest — it is unlikely that peers will agree to adhere to a protocol which does not protect their interests, and a break from protocol will effectively destroy any social contract being formed between peers.

In abstract terms, a protocol $\mathbb{P}$ should specify an interaction model $\mathcal{M}$ which maximises the number of possible worlds described by a system state $\mathbb{S}_i$ in any transition relation $(\mathbb{S}_i, E, \mathbb{S}_f) \in \mathcal{M}$ such that it is almost always possible to find an interaction I modelled by $\mathcal{M}$ which can be performed from a given initial system state description (in question is whether that particular I is desirable to a given agent).

Given that we are concerned here with the *initial* interaction state, wherein no part of the interaction has yet transpired, it should be possible to select an interaction state $\mathcal{S} = (H, \Delta, \mathcal{M})$ which models the system state $\mathbb{S}$ without knowledge of the history of $\mathbb{S}$. Given that $(\mathbb{S}, \emptyset, \mathbb{S}) \in h(\mathbb{S})$, it should be possible to find a system state $\mathbb{S}_i \sim \mathbb{S}$ referred to in $\mathcal{M}$ such that given $H = \emptyset$, for every $F \in \Delta$, it is the case that $(\emptyset \cup F)$ is modelled by $\mathcal{M}$ in accordance with Definition 2.7.

Example 2.7 *Let us return to the premise of Example 2.1, wherein Alanna desires access to the much-vaunted library. Now recall the protocol first displayed in Example 1.1. This protocol* acquire_access *specifies a process by which an agent can obtain access to a given resource, provided that the advocacy of an existing patron of that resource can be obtained. By Definition 2.11:*

$$\mathsf{models}(\mathbb{S}_0, \mathcal{S}_0) \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{selection}(\mathsf{alanna}, \mathbb{S}_0, \mathsf{acquire\_access}) & \wedge \\ \mathsf{initial\_state}(\mathsf{acquire\_access}, \mathcal{S}_0) & \end{array} \right)$$

acquire_access *provides a desirable outcome (in which Alanna becomes able to access her desired information) and is compatible with system state* $\mathbb{S}_0$ *as defined in Example 2.2 provided that Alanna adopts the role of* applicant(library) *such that Applicant =*

alanna, *Resource* = library *and Advocate* = benjamin. *The initial interaction state $\mathcal{S}_0$ is as in Example 2.2 also (albeit with the possible events mapped to messages prescribed by* acquire_access*).*

One part of the contribution of the Open Knowledge project was a service for publishing and indexing LCC protocols. A mechanism for LCC protocol synthesis was also examined in [McGinnis et al., 2005]. As yet however, it has generally been assumed that humans will be primarily be responsible for actual selection of protocols, though work on automatic verification of properties of MAP / LCC protocols has been conducted [Walton, 2004b, Osman, 2007] which could prove to be the basis for autonomous protocol selection by agents.

2.3.2 Transitions of Interaction State

Whilst the system state is only partially accessible to and only partially under the control of the agents within it, the interaction state is something the agents have full control over (within the confines of an accepted interaction protocol). Instead of actually specifying the transition of system states then, we instead simply transition the interaction state, and in doing so enact the actions prescribed by the protocol for such transitions. Those actions can be relied upon to influence the system state accordingly, and maintain the link between interaction state and system state modulo any external events.

An interaction state transition is the combination of (parallel) transitions of the local interaction states accorded to individual agents involved in an interaction:

Definition 2.12 *An interaction dialogue I is the set of messages exchanged between a group of agents Σ driving a transition from an initial interaction state $\mathcal{S}_i$ to a final interaction state $\mathcal{S}_f$, which can be partitioned into a set of local interaction states for each agent $\sigma \in \Sigma$:*

$$\mathsf{transition}(\mathcal{S}_i, I, \mathcal{S}_f) \quad \leftrightarrow \quad \mathsf{actors}(\mathcal{S}_f, \Sigma) \wedge$$

$$\left(I = \bigcup_{\sigma \in \Sigma} M_{(i,\sigma)} = \bigcup_{\sigma \in \Sigma} M_{(o,\sigma)} \;\middle|\; \begin{array}{ll} \mathsf{local\_state}(\mathcal{S}_i, \sigma, \mathcal{S}[\sigma]_i) & \wedge \\ \mathsf{transition}(\mathcal{S}[\sigma]_i, M_{(i,\sigma)}, M_{(o,\sigma)}, \mathcal{S}[\sigma]_f) & \wedge \\ \mathsf{local\_state}(\mathcal{S}_f, \sigma, \mathcal{S}[\sigma]_f) & \end{array} \right)$$

Where:

- $\mathsf{actors}(\mathcal{S}, \Sigma)$ *is true if Σ is the set of agents involved in the interaction described by interaction state $\mathcal{S}$.*

- $\mathsf{local\_state}(\mathcal{S}, \sigma, \mathcal{S}[\sigma])$ *is true if* $\mathcal{S}[\sigma]$ *is the the portion of interaction state* $\mathcal{S}$ *pertaining directly to the agent* σ.

- $\mathsf{transition}(\mathcal{S}[\sigma]_i, M_i, M_o, \mathcal{S}[\sigma]_f)$ *is true if the local interaction state* $\mathcal{S}[\sigma]_i$ *of agent* σ *can transition into state* $\mathcal{S}[\sigma]_f$ *given the reception of the messages in set* M_i*, responding with the messages in set* M_o.

In general, one would expect that there will be one agent which is evidently involved in the interaction from the beginning (being the initiating agent), and that other agents will be introduced as interaction progresses. In the specification above, it might appear that we expect all agents that will be involved in an interaction to be known from the start. This is in fact not the case — we expect that the full set Σ of peers will remain not fully instantiated for a significant part of the duration of an interaction, and that it will not be possible to evaluate all agent transitions immediately. In particular, the set of messages received by an agent must first be dispatched by its peers before they can be responded to, meaning that most agent transitions will only be able to be evaluated incrementally, with the ability to progress the interaction being intermittent between agents. Naturally if interaction is to continue towards completion, it must always be possible to advance at least one peer's portion of the interaction state at any particular point in the interactive process.

Example 2.8 *Alanna wants to use the protocol* acquire_access *to transition from a system state in which she cannot access the library to one in which she can. At this point, we know that Alanna is going to be one of the actors in the final dialogue, but as yet we are uncertain as to which other agents will be involved. The part of the interaction local to Alanna is described by the predicate* transition*:*

$$\mathsf{transition}(\mathcal{S}[\mathsf{alanna}]_i, M_{(\mathsf{alanna},i)}, M_{(\mathsf{alanna},o)}, \mathcal{S}[\mathsf{alanna}]_f)$$

Wherein $\mathcal{S}[\mathsf{alanna}]$ *is the interaction state* $\mathcal{S}$ *known to Alanna, and* $M_{(\mathsf{alanna},i)}$ *and* $M_{(\mathsf{alanna},o)}$ *are the sets of messages received and dispatched during interaction respectively. Naturally, we do not know the content of these two sets from the outset of dialogue either.*

As interaction progresses however, we are able to incrementally instantiate our model. Say that Alanna requests Benjamin's aid as expected. We know then that $\mathsf{request} \in M_{(\mathsf{alanna},o)}$*. We also now know that Benjamin is a participant in the interaction, in the role of* advocate(library)*:*

$$\mathsf{transition}(\mathcal{S}[\mathsf{benjamin}]_i, M_{(\mathsf{benjamin},i)}, M_{(\mathsf{benjamin},o)}, \mathcal{S}[\mathsf{benjamin}]_f)$$

In addition, we know that $\mathsf{request} \in M_{(\mathsf{benjamin},i)}$ *(this is verified, in a circuitous way, by the requirement that the set of all dispatched messages matches the set of all received messages, as well as the requirement in Definition 2.14 later that all messages are used to progress interaction).*

Finally, if the resulting dialogue does indeed unfold as described in Example 2.1, we shall be able to introduce Charlotte to the interaction, and fully instantiate our model such that:

$$\mathsf{transition}(\mathcal{S}\,[\mathsf{alanna}]_i, \{permit\}, \{request\}, \mathcal{S}\,[\mathsf{alanna}]_f)$$

$$\mathsf{transition}(\mathcal{S}\,[\mathsf{benjamin}]_i, \{request\}, \{advocate(alanna)\}, \mathcal{S}\,[\mathsf{benjamin}]_f)$$

$$\mathsf{transition}(\mathcal{S}\,[\mathsf{charlotte}]_i, \{advocate(alanna)\}, \{permit\}, \mathcal{S}\,[\mathsf{charlotte}]_f)$$

At which point we can infer that interaction $\imath = \{request \prec advocate(alanna) \prec permit\}$.

Other than the initiating agent, agents are generally inducted into an interaction because they have been identified as satisfying some logical requirement described by the interaction protocol and are then sent a message to that effect; in some systems however, peers are selected in advance of interaction. Either way, selection of peers may involve invocation of a matchmaking service in order to identify suitable peers, such as that specified by [Lambert and Robertson, 2005].

2.3.2.1 Decomposing the Interaction State

The interaction state for a distributed interaction supported by an interaction protocol can be seen to consist of three parts; the protocol itself, the active state and the normative model. The protocol is a distributed interaction model as described by Definition 2.8 — in practical terms, it consists of a set of process models which can be assigned to agents acting in particular roles in a compatible interaction. The set of active clauses is the set of process models *in situ* — that is, it consists of copies of process models extracted from the protocol which have been adopted by agents, and which are in some state of execution (which is represented by transformations of the models, as will be described below). The normative model describes any social norms applicable to the interaction. We do not concern ourselves much with norms — basically the normative model in a interaction state is a black box for the elements of social behaviour which we do not address in this thesis. For example, the normative model might provide rules

restricting the actions an agent can take whilst engaged in an interaction (e.g. to ensure that the agent does not sabotage the integrity of the interaction by simultaneously engaging in another interaction which changes the underlying basis for the first interaction), or provide rules restricting how long an agent can stall in providing a response to a message (e.g. in order to acquire information to help it resolve a constraint without resorting to abduction). LCC provides a simple example of a normative element; a protocol designer can add 'common knowledge' predicates to a protocol which provide a standard means to resolve a particular logical constraint.[6]

In a distributed system, the interaction state for an interaction must be distributed amongst all interacting peers. In question then is how much information each agent requires to conduct its role(s) in interaction. At the very least, each agent needs access to the process models for every role it adopts. Since these models are extracted from a common interaction protocol, and agents may adopt additional roles as interaction progresses, it would seem expedient for every agent to possess a full copy of the protocol. This confers an additional benefit — individual peers will always be aware of the provenance of certain events (because they can determine the pre-conditions for the dispatch of a given message by any of their peers in the interaction from the process models for their roles) and thus will be able to verify from events observed the the decisions peers must have taken in their given roles and what commitments they essentially make if they are to continue to adhere to the common protocol.

Definition 2.13 *The local interaction state $\mathcal{S}[\sigma]$ for an agent σ is a partition of the interaction state $\mathcal{S}$ containing the protocol $\mathbb{P}$, the normative model $\mathcal{N}$ and the subset $\mathcal{M}[\sigma]$ of active clauses adopted by σ:*

$$\mathsf{local\_state}(\mathcal{S}, \sigma, \mathcal{S}[\sigma]) \leftarrow \left(\begin{array}{ll} \mathsf{protocol}(\mathcal{S}, \mathbb{P}) & \wedge \\ \mathsf{norms}(\mathcal{S}, \mathcal{N}) & \wedge \\ \mathsf{role\_models}(\mathcal{S}, \sigma, \mathcal{M}[\sigma]) & \wedge \\ \mathcal{S}[\sigma] = \mathsf{peer\_state}(\mathcal{M}[\sigma], \mathbb{P}, \mathcal{N}) & \end{array} \right)$$

Where:

- $\mathsf{protocol}(\mathcal{S}, \mathbb{P})$ *is true if $\mathbb{P}$ is the protocol to which interaction state $\mathcal{S}$ adheres.*
- $\mathsf{norms}(\mathcal{S}, \mathcal{N})$ *is true if $\mathcal{N}$ describes the social norms to which interaction state $\mathcal{S}$ adheres.*

[6] This does not invalidate our own contribution however, since these predicates tend only to decompose a constraint into smaller, more elementary constraints, which are still subject to interpretation by agents.

- role_models($\mathcal{S}, \sigma, \mathcal{M}[\sigma]$) *is true if* $\mathcal{M}[\sigma]$ *is the set of process models adopted by agent* σ *so far according to interaction state* $\mathcal{S}$*. For each role R adopted by* σ*, there exists a process model* $\mathcal{M}[R, \sigma] \in \mathcal{M}[\sigma]$.

- peer_state($\mathcal{M}[\sigma], \mathbb{P}, \mathcal{N}$) *returns the local interaction state* $\mathcal{S}[\sigma]$ *collectively described by process models* $\mathcal{M}[\sigma]$*, protocol* $\mathbb{P}$ *and social norms* $\mathcal{N}$.

Selection of roles in interaction, and thus the process models to use from the interaction protocol, depends on the state of interaction. At the start of interaction, the agent responsible for initiating the interaction and selecting its protocol must select an initial role. Clearly this must be a role from which it can immediately transition its state and generate events which will induct further peers into the interaction (as defined immediately below). Aside from this requirement however, an agent is free to select any role which suits its needs, and we do not concern ourselves with this further. Other agents are essentially drawn into roles by the events they observe; they can only select a role which allows them to respond to such events, otherwise the interaction will not proceed. As interaction progresses, the process models described within an agent's local interaction state will 'unfold',[7] essentially closing off outcomes which no longer apply to the current interaction state, and opening paths which are admissible given the events observed.

2.3.2.2 Local State Transition

The actual conduct of interaction is in the local state transitions made by agents in response to observations, and the effect such transitions have on an agent, its peers and its environment.

Definition 2.14 *The local interaction state* $\mathcal{S}[\sigma]$ *of an agent* σ *cannot advance without an event to trigger that advancement. This forms a base case for local state transition:*

$$\text{transition}(\mathcal{S}[\sigma], \emptyset, \emptyset, \mathcal{S}[\sigma]) \quad \leftarrow \quad \text{true}$$

Otherwise, an agent σ *will dispatch a set of messages* $(M_o \cup M_n)$ *in response to receiving messages* M_i *as it transitions from a local interaction state* $\mathcal{S}[\sigma]_i$ *to a state* $\mathcal{S}[\sigma]_f$*:*

$$\text{transition}(\mathcal{S}[\sigma]_i, M_i, (M_o \cup M_n), \mathcal{S}[\sigma]_f) \quad \leftarrow$$

[7] Imagine a sheet of paper, folded several times. Large parts of its surface are hidden, and the observer can only guess what is on it. As the sheet gets unfolded however, more of its hidden surface is revealed. Eventually, all of its surface will become clearly apparent. This would appear to be the reasoning behind the notion of 'unfolding' a logical clause.

$$\left(\begin{array}{ll} \mathsf{role\_model}(\mathcal{S}[\sigma]_i, \mathcal{M}[R,\sigma]_i) & \wedge \\ \mathcal{M}[R,\sigma]_i \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_n} \mathcal{M}[R,\sigma]_j & \wedge \\ \mathsf{updated\_state}(\mathcal{S}[\sigma]_i, \mathcal{M}[R,\sigma]_j, \mathcal{S}[\sigma]_j) & \wedge \\ \mathsf{transition}(\mathcal{S}[\sigma]_j, M_j, M_o, \mathcal{S}[\sigma]_f) & \end{array}\right)$$

Where:

- $\mathsf{role\_model}(\mathcal{S}[\sigma], \mathcal{M}[R,\sigma])$ *is true if* $\mathcal{M}[R,\sigma]$ *is a role model describing the requirements placed on agent* σ *when acting in role R given the local interaction state* $\mathcal{S}[\sigma]$.

- $\mathcal{M}[R,\sigma]_i \xrightarrow{M_i,\mathcal{S}[\sigma],M_j,M_n} \mathcal{M}[R,\sigma]_j$ *is an unfolding by agent* σ *of process model* $\mathcal{M}[R,\sigma]_i$ *into* $\mathcal{M}[R,\sigma]_j$ *in response to receiving messages* (M_i/M_j) *in accordance with the protocol stored within local interaction state* $\mathcal{S}[\sigma]$*; this act of unfolding generates the set of messages* M_n.

- $\mathsf{updated\_state}(\mathcal{S}[\sigma]_i, \mathcal{M}[R,\sigma], \mathcal{S}[\sigma]_j)$ *is true if* $\mathcal{S}[\sigma]_j$ *describes the local interaction state after re-integration of process model* $\mathcal{M}[R,\sigma]$ *into prior local interaction state* $\mathcal{S}[\sigma]_i$.

- $\mathsf{transition}(\mathcal{S}[\sigma]_i, M_i, M_o, \mathcal{S}[\sigma]_f)$ *is true if agent* σ *can transition from local interaction state* $\mathcal{S}[\sigma]_i$ *to state* $\mathcal{S}[\sigma]_f$ *given the reception of messages* M_i*, dispatching the set of messages* M_o *in response.*

Given a distributed dialogue protocol, an agent can transition state by successfully unfolding the process model for any one of its adopted roles in interaction. By executing the actions prescribed within, an agent will generate events which will change the system state and drive interaction further. A process model can only be unfolded given the satisfaction of any conditions imposed by the model however. Logical constraints require an evaluation of agent beliefs, which are derived from observation of events and personal introspection. Meanwhile, agents can only take certain actions if certain defined messages have been received from peers during interaction; these messages compose the actual interaction dialogue proper. The connection between interaction state and system state is maintained by the evaluation of constraints and execution of responses; the imposition of constraints ensures that the system state has bearing on the interaction state, and the execution of actions (including the generation of new dialogue) ensures that the system state changes alongside the interaction state.

At this point, we can demonstrate how a role model drawn from an LCC protocol can be unfolded; this is done incrementally, where in each step an agent either dispatches a message, formally receives a message, satisfies a constraint or performs an action. Unfolding is performed by finding a compatible rewrite rule and, subject to any conditions, applying it to the role model. Whilst we concentrate on LCC specifically, one can imagine an analogous process for any equivalent agent protocol language.

Definition 2.15 *A role model* $\mathcal{M}\ [R,\sigma] = \mathsf{a}(R,\sigma) :: P$ *drawn from an LCC protocol* $\mathbb{P}$ *can be incrementally unfolded by applying appropriate* **rewrite rules** *to* $\mathsf{a}(R,\sigma) :: P$ *and its sub-clauses:*[8]

$$\mathsf{a}(R_n,\sigma) :: P \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} \mathsf{a}(R_n,I) :: E \quad \textit{if} \quad P \xrightarrow{R_n,M_i,M_j,\mathcal{S}[\sigma],M_o} E$$

$$P_1 \ \mathsf{else}\ P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E \quad \textit{if} \quad P_1 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E \ \vee \ P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E$$

$$P_1 \ \mathsf{par}\ P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E_i \quad \textit{if} \quad \left(\begin{array}{ll} P_1 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E & \wedge \\ \mathsf{interleave}(E,P_2,E_i) & \end{array}\right) \vee \left(\begin{array}{ll} P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E & \wedge \\ \mathsf{interleave}(P_1,E,E_i) & \end{array}\right)$$

$$P_1 \ \mathsf{then}\ P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} P_1 \ \mathsf{then}\ E \quad \textit{if} \quad \mathsf{closed}(P_1) \wedge P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E$$

$$P_1 \ \mathsf{then}\ P_2 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E \ \mathsf{then}\ P_2 \quad \textit{if} \quad P_1 \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E$$

$$M \Rightarrow \mathsf{a}(R_r,\sigma_r) \xrightarrow{R,M_i,M_i,\mathcal{S}[\sigma],\{\mathsf{m}(\mathsf{a}(R,\sigma),\mathsf{a}(R_r,\sigma_r),M)\}} \mathsf{c}(M \Rightarrow \mathsf{a}(R_r,\sigma_r))$$

$$P \leftarrow M \Leftarrow \mathsf{a}(R_s,\sigma_s) \xrightarrow{R,M_i,M_i/\mathsf{m}(\mathsf{a}(R_s.\sigma_s),\mathsf{a}(R,\sigma),M),\mathcal{S}[\sigma],\emptyset} \mathsf{c}(M \Leftarrow \mathsf{a}(R_s,\sigma_s)) \ \mathsf{then}\ P \quad \textit{if} \quad \mathsf{m}(\mathsf{a}(R_s,\sigma_s),\mathsf{a}(R,\sigma),M) \in M_i$$

$$P \leftarrow C \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E \quad \textit{if} \quad \mathsf{satisfied}(\sigma,C) \wedge P \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} E$$

$$\mathsf{a}(R_n,\sigma) \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} \mathsf{a}(R_n,\sigma) :: P \quad \textit{if} \quad \mathsf{role\_model}(\mathcal{S}[\sigma],\mathsf{a}(R_n,\sigma) :: P)$$

$$A \xrightarrow{R,M_i,M_j,\mathcal{S}[\sigma],M_o} \mathsf{c}(A) \quad \textit{if} \quad \mathsf{execute}(\sigma,A)$$

A role model $\mathsf{a}(R,\sigma) :: P$ *is* **closed** *(i.e. role R has been performed by* σ *in its entirety), if all of its sub-clauses are closed. This can be determined using the following rules:*

[8] For brevity, we have omitted a few rewrite rules which are merely pedantic variations on those present.

$$\begin{aligned} \mathsf{closed}(\mathsf{a}(R,\sigma) :: P) &\leftarrow \mathsf{closed}(P) \\ \mathsf{closed}(P_1 \text{ then } P_2) &\leftarrow \mathsf{closed}(P_1) \wedge \mathsf{closed}(P_2) \\ \mathsf{closed}(\mathsf{c}(P)) &\leftarrow \mathsf{true} \end{aligned}$$

Individual rewrite rules may be conditional on the satisfaction of certain logical propositions:

- $\mathsf{interleave}(P_1, P_2, E)$ *produces an interleaving* E *of parallel procedures* P_1 *and* P_2 *such that any closed sub-clauses of either procedure are sequentially ordered, according to the following (strictly ordered) rules:*

$$\begin{aligned} \mathsf{interleave}(P_1, P_2, P_1 \text{ then } P_2) &\leftarrow \mathsf{closed}(P_1) \\ \mathsf{interleave}(P_1, P_2, P_2 \text{ then } P_1) &\leftarrow \mathsf{closed}(P_2) \\ \mathsf{interleave}(P_1 \text{ then } P_2, P_3, P_1 \text{ then } E) &\leftarrow \mathsf{closed}(P_1) \wedge \mathsf{interleave}(P_2, P_3, E) \\ \mathsf{interleave}(P_1, P_2 \text{ then } P_3, P_2 \text{ then } E) &\leftarrow \mathsf{closed}(P_2) \wedge \mathsf{interleave}(P_1, P_3, E) \\ \mathsf{interleave}(P_1, P_2, P_1 \text{ par } P_2) &\leftarrow \mathsf{true} \end{aligned}$$

  This ensures that a role model with parallelised sub-procedures will describe the order in which events are handled when unfolded.

- $\mathsf{satisfied}(\sigma, C)$ *is true if logical constraint* C *can be satisfied using the beliefs of agent* σ*.*

- $\mathsf{role\_model}(\mathcal{S}[\sigma], \mathsf{a}(R,\sigma) :: P)$ *is true if there exists a role model* $\mathsf{a}(R,\sigma) :: P$ *defined within* $\mathbb{P}$*, the protocol used to model* $\mathcal{S}$ *(*$\mathbb{P}$ *is shared by all agents involved in* $\mathcal{S}$ *and thus is accessible within* $\mathcal{S}[\sigma]$*). This is identical in principle to* role_model *as defined in Definition 2.14.*

- $\mathsf{execute}(\sigma, A)$ *calls upon agent* σ *to execute action* A *immediately. This may affect a change in agent state, or a change in the environment. There a few special actions worth noting:*

  - null *does nothing. It is often convenient to perform a null action — for example, within the base cases of recursively-defined role models.*
  - succeed *indicates that an agent has achieved its goal within the interaction; the idea is that an agent can enact any plans contingent upon achieving a desired outcome upon encountering this action (e.g. accessing the data within a library).*

– fail *indicates that an agent has not achieved its goal within the interaction; the idea is that if an agent has any alternative plans in case of failure, it can enact them upon encountering this action. Note that* fail *does* not *indicate a breakdown of the interaction itself — instead, it merely indicates that an agent's ideal outcome has not emerged (e.g. an agent's request is refused).*

The above rewrite rules are not quite the same as those found in [Robertson, 2004, Robertson et al., 2008] — this is to better accommodate our contribution. In this case, the grammar for LCC has been made slightly more permissive, and a stronger distinction is made between *declarative* constraints (which is what we usually mean by the term 'logical constraint') and *procedural* constraints (which we prefer to refer to as actions). Traditionally LCC does not make a strong distinction, and allows both active and reactive constraints which may or may not exhibit a procedural effect when evaluated.[9] Because our contribution is only concerned with investigating the truth of logical propositions, and not with the side-effects of evaluating propositions which are merely wrappers for procedures, we want to make it as easy as possible for our portrayal mechanism to be able to distinguish between the two.

Example 2.9 *By applying the rewrite rules of Definition 2.15, Alanna can transform her process model, driving her interaction with Benjamin and Charlotte forward. For instance . . .*

> *Alanna should request help from Benjamin if the library is not accessible and Benjamin is a patron of the library . . .*
>
> a(applicant(library), alanna) ::
> request ⇒ a(advocate(library), benjamin)
> ⇐ ¬ accessible(alanna, library) ∧ patron(benjamin, library) then . . .

. . . becomes . . .

> *Alanna has requested help from Benjamin . . .*
>
> a(applicant(library), alanna) ::
> c(request ⇒ a(advocate(library), benjamin)) then . . .

*. . . because Alanna is satisfied that the library is inaccessible (*accessible(alanna, library) *is false) and that Benjamin is a patron of the library (*patron(benjamin, library) *is true). As a side effect, this produces message* request *directed towards Benjamin in*

[9] In this respect, LCC treats propositions as if predicate calls in Prolog, revealing its roots as a prototype written in that language.

the role of advocate(library)*, which allows us to enter Benjamin into the interaction as described in Example 2.8.*

Assume that Benjamin responds to Alanna's request, and recommends her to Charlotte. Charlotte, deeming Alanna eligible, then dispatches a message permit *to Alanna. This allows her to unfold her model further:*

a(applicant(library), alanna) ::
  c(request ⇒ a(advocate(library), benjamin)) then
  c(permit ⇐ a(controller(library), charlotte)) then
  succeed ← accessible(alanna, library).

At this point, all that remains is to complete the interaction by confirming that the library is indeed now accessible (which Alanna will either accept as a consequence of receiving Charlotte's permission, or will confirm independently).

2.3.3 Completing Interaction

Interaction is complete ('closed') once all agents have completed their roles in the interaction, and thus no further events are expected to be observed pertaining to that interaction:

Definition 2.16 *An interaction state $\mathcal{S}$ models a system state $\mathbb{S}$ if $\mathcal{S}$ is the final state of an interaction $\mathcal{I}$ which is part of the recent history of $\mathbb{S}$:*

$$\mathsf{models}(\mathbb{S},\mathcal{S}) \quad \leftarrow \quad \exists!\,\mathcal{I}\,.\,(\,\mathsf{interaction}(\mathcal{S},\mathcal{I}) \wedge \mathsf{stimulus}(\mathbb{S},\mathcal{I})\,)$$

Where:

- $\mathsf{interaction}(\mathcal{S},\mathcal{I})$ *is true if interaction state* $\mathcal{S} = (H,\Delta,\mathcal{M})$ *as per Definition 2.5 such that interaction* $\mathcal{I} = (H \cup F)$ *for some* $F \in \Delta$ *(i.e. interaction* $\mathcal{I}$ *is one of the interactions which* $\mathcal{S}$ *might describe).*

- $\mathsf{stimulus}(\mathbb{S},\mathcal{I})$ *is true if* $(\mathbb{S}_i,\mathcal{I},\mathbb{S}) \in h(\mathbb{S})$ *for some system state* $\mathbb{S}_i$ *as per Definition 2.6 (i.e. interaction* $\mathcal{I}$ *is a stimulus for the transition of the system into state* $\mathbb{S}$*).*

Given that a final interaction state $\mathcal{S} = (\mathcal{I},\emptyset,\mathcal{M})$ such that it describes a unique interaction $\mathcal{I}$ which has finished, with no further events due, it is clear that $\mathcal{S}$ models any system state $\mathbb{S}$ which immediately transpires at the end of $\mathcal{I}$. Note that in our specification, the interaction state will always model the system state immediately after the end of interaction because that system state will be a product of the actions performed in

transitioning the interaction state to its final form (along with any coincident external events which failed to stop interaction).

If the interaction state is inferred to model the system state by means of Definition 2.16, then the interaction has clearly ended (otherwise it would not already be in the history of the system state). However, there are other means to infer that the interaction state models the system state without requiring interaction to be complete. Therefore in Definition 2.10, we perform an additional check ($\mathsf{closed}(\mathcal{S})$) in order to ensure that interaction has actually finished.

2.4 Distributed Decision Making

Within a distributed system, each agent has its own view of the system — an agent extrapolates the system state from its own personal state and its observations of its environment. If an agent cannot access the personal states of its peers and it cannot directly perceive in its entirety the environment in which the agent system is situated, then we must accept that the agent will not be able to construct a complete description of the system state. This does not preclude it from partaking in interaction of course — aside from the coordination of behaviour, interaction is often used to gather information from peers. Agents can infer more details about a system from the behaviour of their peers, particularly if peers can be expected to adhere to some known interaction model.

In Definition 2.3, we specified that an interaction could be understood by the state transitions caused by the events comprising that interaction. A given interaction model is unlikely to actually define transitions from every possible concrete system state to every other concrete system state. Instead, it would define abstract states which could describe any number of actual system states based on what aspects of the system actually have bearing on the modelled interaction. To be able to effectively apply an interaction model to a new interaction, it is only necessary that agents know enough about the system to be able to respond to events.

As an interaction unfolds, agents can refine their understanding of the system by observing their peers' actions, and comparing them with the interaction model. Assuming adherence to the model, the observation of a given event indicates a particular system transition; agents can thus infer that the real system state lies in the intersection of possible worlds described by the transition relation in the interaction model, and their own system state description. Of course, this assumes that agents always

perceive the world accurately.

The basic justification for non-deductive reasoning is the need to make decisions with incomplete information. Consider an interaction wherein an agent must select one of two actions, each of which leads to a different system state, both of which are justifiable given the agent's knowledge of the system. Based on a balance of admissibility, probability and risk, the agent assumes a given system state and selects an action — and indeed, continues to act based on that assumption. Any peers observing the actions of that agent will then be given cause to infer that assumption themselves, regardless of whether or not it accurately reflects reality.

Eventually, two agents with mutually inconsistent views of the world are likely to interact, and this is going to affect how the agents behave. An event will occur which cannot be reconciled with the state an agent believes the system to be in, and the interaction state will no longer model that system state, compelling the agent to resort to one of the options described at the end of §2.2.1. However because the perception that the interaction state does not model the system state is not universal, it may be that such efforts actually lead to dissonance between interaction and system state from the perspective of the other agent. In fact, it may not be possible for the interaction state to model all views of the system simultaneously at all. Note that this can occur even if all agents are acting in adherence to a common protocol.

Example 2.10 *In our running example, Alanna believes that she can be trusted with access to the library. It is possible however, that Benjamin in fact believes that Alanna is not trustworthy. Assume that dialogue is conducted in adherence to protocol* acquire_access *as per Example 2.7; it is commonly known that an advocate will recommend an applicant if* trustworthy(*Applicant*, *Resource*). *Thus we have a scenario wherein Alanna expects Benjamin to recommend her to Charlotte, but Benjamin will not, due to contradictory beliefs. This means that in spite of having a common protocol, there will still come a point in which the interaction state will appear not to model the system state.*

Let us look at another less admittedly subjective constraint. A controller will only permit access if the advocate is already a trusted peer and eligible(*Applicant*, *Resource*). *It may be that Charlotte lacks information about Alanna, or even has false beliefs about her which would lead her to infer that Alanna is ineligible. Again, if Alanna believes that she* is *eligible, this will create a circumstance where interaction state does not model apparent system state.*

This creates a dilemma. The interaction has already gone awry (from the perspective of at least one agent), but all peers are committed to completing the interaction according to the protocol. This may force agents to act in ways counter to their best judgement in order to fulfil a social contract (not so evident in the above example, but imagine if Benjamin trusts Alanna, but *Charlotte* does not — by the protocol, Charlotte must grant access if Alanna is eligible anyway, because the evaluation of trustworthiness is conferred to Benjamin alone). Another possibly more damaging scenario may arise; an agent may think that events support their perception of the system state, but in fact are indicative of another state which is actually less beneficial to them, such that the agent actually acts against its own best interests (e.g. Alanna is *not* trustworthy, but Benjamin and Charlotte assume that she is, indirectly giving her access to a sensitive resource). We would like to prevent either scenario from occurring where possible.

It might be worth noting that technically there is a very easy way to ensure consistency between different views of the system on the part of individual agents. That is simply by applying epistemic modal logic [Fagin et al., 1995] — if beliefs are explicitly accorded to individual agents, then there is no inconsistency (because whilst 'Alanna is eligible' and 'Alanna is not eligible' are inconsistent statements, 'Alanna *believes* that Alanna is eligible' and 'Charlotte *believes* that Alanna is not eligible' are not). However, this does not in practice actually solve the problems just described, because essentially all it does is divest agents of responsibility for peers' beliefs, when what we actually want is for agents to be in agreement. We are more interested in the possibility that if agents were able to combine their pertinent beliefs, they would actually be able to draw better conclusions.

Our preferred approach then is to have a mechanism to repair system state descriptions, or more specifically, the beliefs of agents used to make decisions during interaction.

Example 2.11 *Assume that Alanna believes that she is trustworthy, but Charlotte believes otherwise. It is Benjamin who will ultimately have to decide one way or the other. We want to be able to identify the conflict before Benjamin makes his decision and, if possible, resolve it. We do not however want to sift through everything that Alanna and Charlotte believe about the system, because most of it is not relevant to the decision to be made.*

What we would like is for Benjamin, prior to evaluating the proposition trustworthy(alanna, library)*, to invite Alanna and Charlotte to posit their expectations:*

Alanna: *true* $\rightarrow$ trustworthy(alanna, library).
Charlotte: trustworthy(alanna, library) $\rightarrow$ *false.*

Noting a dispute, Benjamin would then permit agents to elaborate upon their claims, or to attack one another's assumptions. Assume that Charlotte's distrust of Alanna is based on an erroneous belief about past behaviour:

Charlotte: *true* $\rightarrow$ stole_data(alanna, ...) ...

This might be immediately quashed by Benjamin, who knows that Alanna did not do such a thing. Assuming that Charlotte cannot produce an alternative argument, it would then appear that it is admissible to trust Alanna, and upon resuming the main interaction, there would be no apparent dissonance between the interaction state and the apparent system state.

In essence, we want to allow agents to share information, such that fewer unfounded assumptions are made (because the system state descriptions agent use are more 'complete') and the outcome of interaction meets the expectations of peers (because inconsistencies between system state descriptions have been identified and removed). It could be said that what we are looking for is a distributed truth maintenance system [Huhns and Bridgeland, 1991] — albeit one which is specifically targeted at making specific multi-agent coordination tasks more robust, rather than ensuring global consistency of beliefs (which we argue in the next chapter to be prohibitively expensive and impractical computationally).

We want to do this however in a fashion which does not negate the advantage of using interaction protocols (compact descriptions of interaction with well-defined events and state transitions). Therefore, we want to frame this as a distinct interactive process which works alongside the main interaction, and merely intercedes where necessary to ensure that the most rational outcome arises. This has the benefit of allowing us to use unaugmented protocols. In Chapter 5, we shall specify such a process. Prior to this however, we shall demonstrate how argumentation can be used to repair inconsistencies between agent beliefs, giving us a formal basis upon which to build our ultimate contribution.

In summary — in this chapter we formalised our preferred notion of interaction and described how interaction between intelligent agents can be identified and guided by various means, but in particular using dialogue protocols written in the Lightweight Coordination Calculus [Robertson, 2004]. We provided a model for the process of executing an interaction in accordance with an LCC protocol, so as to provide a framework

upon which we can later attach our contribution (to be specified in Chapter 5). We also considered the problems inherent in interaction between independent peers even when guided by a protocol, with particular emphasis this last section on contrasting agent beliefs — this sets up the motivation for the next chapter. What we shall do in Chapter 3 then is build up a case for using argumentation to reconcile the beliefs of agents prior to decisions being made, in the hope that this leads to better outcomes for multi-agent interaction.

Chapter 3

An Argumentative Approach to Defeasible Reasoning

It may be that an agent is unable to rely entirely on deductive reasoning in order to function in a sufficiently complex environment. It may find it necessary instead to make decisions based on assumptions drawn from untested hypotheses. By making such assumptions, an agent can act in uncertain and volatile circumstances, rather than succumb to decision paralysis. It must always be recognised however that any assumptions made, even when consistent with all evidence available at the time, may later prove to be false. Thus, an agent must be willing to revise its beliefs and discard prior conclusions upon the discovery of new information.

In a multi-agent system made up of heterogeneous agents, it can be expected that the assumptions made will vary from agent to agent, based on each agent's particular experiences and biases. This may lead to a divergence of beliefs, which in turn can lead to even ostensibly cooperative agents responding differently to the same set of circumstances. In particular, an agent might expect that a peer will account for certain factors before making a decision, factors which the peer might actually be ignorant of — such a presumption could adversely affect the outcome of any interaction which happens to be contingent upon such a decision. Consequently, it is important to consider how social mechanisms such as dialogue can be used by agents to periodically compare observations and challenge expectations, ideally to arrive at a unified viewpoint — but if not that, then at least to arrive at a consensus to the effect that conflicting beliefs can still be held to be based on rational interpretations of all the evidence available. Such a consensus hinges upon having a robust model for describing how those conflicting beliefs can be inferred from observed data, and how those beliefs then interact within

a given social context.

We should not make the mistake of assuming however that dialogue is the immediate process from which agents derive their beliefs. Dialogue is merely a means to introduce new hypotheses into the internal reasoning machinery of the listener. Before we can properly consider social reasoning, we must first consider how agents determine beliefs in isolation so that we can then understand the influence of dialogue upon that process.

Argumentation research has provided generic frameworks within which to comprehend the process of defeasible reasoning (for example [Bondarenko et al., 1993, Dung, 1995, Kowalski and Toni, 1996, Kakas et al., 1998]), and such frameworks often play proxy to any of a number of alternative logics ([Kakas and Toni, 1999] mentions default logic [Reiter, 1980], modal logics [McDermott, 1982] and auto-epistemic logic [Moore, 1985] as examples). Interestingly, despite being based on a naturally multi-agent metaphor, argumentation is often distilled in the literature such that it becomes an introspective process which can easily be confined to a single actor[1] — it is this line of research which we most closely follow in this chapter, despite our intention to then distribute that process as an essential part of this thesis' contribution.

In summary, this chapter concerns itself with the use of argumentation to derive theories which agents can then use to make decisions. In §3.1, we consider the fundamental mechanics of theory production, in order to ensure that we have a foundation on which to then build our argumentation system. In §3.2, we specify an assumption-based argumentation system which can produce admissible theories given a set of candidate hypotheses. This will allow us to explore in the next chapter how argumentation can be distributed amongst a group of peers.

3.1 Requirements of Practical Reasoning

Before moving onto the particulars of our model for defeasible reasoning, it is worth identifying some of the fundamental elements of logical, agent-oriented reasoning; such elements serve to provide a useful foundation upon which to build further models. This also provides an opportunity to summarise many of the practical difficulties faced by any agent which needs to reason about and act within a complex environment.

An agent may have available to it any number of information sources. These information sources may be internal to an agent (e.g. its memories, goals or the internal

[1] Not always however — see for example [Rahwan et al., 2003, Prakken, 2005].

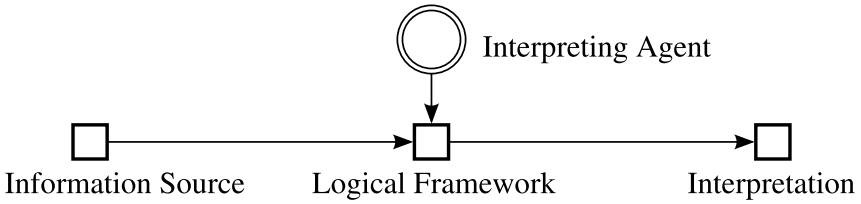


Figure 3.1: *An agent applies a logical framework to interpret some information source.*

state of its components). They may also be external, part of an agent's environment (various artefacts, records, communications from other agents, etc.). In order to interpret such information sources, some kind of *logical framework* is required. Such a framework can be used to deconstruct compatible streams of information into meaningful logical sentences from which can be derived further conclusions by application of some inference procedure:

Definition 3.1 *A* **logical framework** *can be described by a pair* $(\mathcal{L}, \vdash)$ *where:*

- $\mathcal{L}$ *defines all interpretable sentences under this logic framework —* $\mathcal{L}$ *can be treated as the set containing every such sentence to the effect that if an information source* Θ *is interpretable by* $(\mathcal{L}, \vdash)$*, then* $\Theta \subseteq \mathcal{L}$ *(i.e. we can consider* Θ *to be a set of sentences drawn from* $\mathcal{L}$*).*

- $\vdash$ *is an inference procedure used to derive conclusions from sets of premises by application of inference rules encoded within* $\vdash$*. If a conclusion* $\varphi \in \mathcal{L}$ *can be inferred from an information source* $\Theta \subseteq \mathcal{L}$ *using* $\vdash$*, then it can be stated that* $\Theta \vdash \varphi$.

The purpose of a logical framework is to provide syntax for a corpus of information and pragmatics for its interpretation.[2]. Any single information source may be interpreted radically differently depending on the logical framework used, although most artificial information sources are created with a particular syntax (and semantics) in mind. An agent's knowledge base also constitutes an information source, albeit a derivative one. An agent need not apply the same framework to all information sources available to it.

In this work, we concern ourselves primarily with logical frameworks which restrict themselves to only monotonic inference, and is closed under negation.

[2]Strictly speaking a logical framework does *not* provide semantics — the meaning attributed to a body of information cannot be adequately determined without additional context.

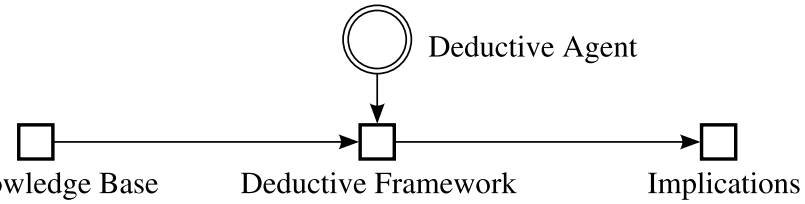


Figure 3.2: *An agent applies a deductive framework to draw conclusions from its knowledge base.*

Definition 3.2 *A* **deductive framework** *is a logical framework* $(\mathcal{L}, \vdash)$ *in which all inference using* $\vdash$ *is monotonic.*[3]

- *Given a sentence* $\varphi \in \mathcal{L}$*, there exists a direct negation* $\neg\varphi \in \mathcal{L}$ *such that* $\varphi \wedge \neg\varphi \vdash$ false.

Ideally, an agent would be able to restrict itself to purely deductive inference, such that provided the information available is accurate, the agent can have perfect confidence in any conclusions inferred. Unfortunately, within a sufficiently complex environment which can be described as *inaccessible*, *dynamic* and *non-deterministic* (amongst other qualities) [Russell and Norvig, 1995], such an idealistic policy may not be sufficient for that agent to achieve its goals. Consider:

Inaccessibility — An inaccessible environment is one in which an agent cannot deduce the environment's complete state. Certain facts about the environment cannot be soundly inferred, and as a result an agent might have to make assumptions based on what it perceives to be the most likely scenario given previous observations.

Dynamism — A dynamic environment's state can change without input from an agent. If an agent is too slow to respond to events, then the world may change, removing the justification for its action. Even if an agent *can* acquire the information it requires to soundly determine the optimal response to an event, it may be impossible to do so within the window of opportunity.

Non-determinism — Often a side-effect of inaccessibility, a non-deterministic environment is one in which there can be more than one possible result of an action,

[3] This is basically the same as the deductive framework $(\mathcal{L}, \mathcal{R})$ used in [Dung et al., 2007] and other papers with the rules in R subsumed within $\vdash$, except that $(\mathcal{L}, \vdash)$ is always closed under negation, which is not presumed by $(\mathcal{L}, \mathcal{R})$.

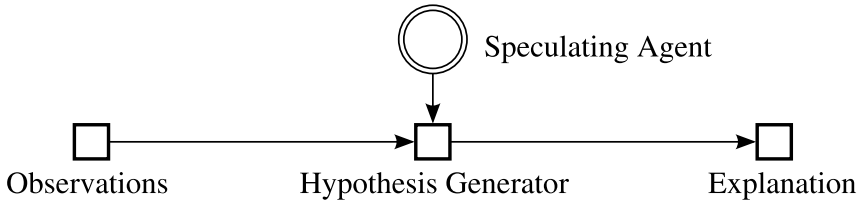


Figure 3.3: *An agent applies a hypothesis generator to provide an explanation for its observations.*

with no apparent factor to indicate one result over another. The deductive agent must consider all disjunctions of possible consequences — something which in most logics is computationally prohibitive (and indeed is often disallowed in many logics to improve computability [Levesque and Brachman, 1987]), without any expectation of a firm final conclusion.[4]

All these factors serve to make deductive reasoning difficult — taken together they make it all but impossible for an agent of limited resources to perform any kind of complex reasoning which does not involve some degree of abductive or inductive inference.

Definition 3.3 *A* **hypothesis generator** *is any logical framework* $(\mathcal{L}, \vdash)$ *in which* $\vdash$ *is used to perform non-monotonic reasoning.*

- *Given an information source* $\Theta \subseteq \mathcal{L}$*, a hypothesis generator can produce a set of hypotheses* H *such that* $\Theta \vdash h$ *for every* $h \in H$*.*

An agent can use a hypothesis generator to produce possible explanations for its observations; it can then choose to assume some of these hypotheses if no contradictory evidence is available and the advantages of acting based on such assumptions is considered to outweigh the inherent risk of being wrong. This can be done by using a deductive framework *within* a hypothesis generator to perform deductions with candidate assumptions — if a contradiction can be generated within a given subset of assumptions, then the hypothesis generator should not generate those assumptions together in the same context, but should instead find a different set of assumptions which provide a consistent theory.

An agent's beliefs can be said then to be based on a balance of empirical observation and hypothetical speculation as befits the environment in which the agent finds

[4]This tends to motivate the use of probabilistic methods to choose the most likely result. Tacit assumptions are still being made however, and so such methods are intrinsically non-monotonic.

itself. These beliefs can then be used to made decisions and solve problems encountered in an agent's pursuit of its goals. However, decision-making based on a set of beliefs is only rational if those beliefs are *consistent* with one another.

Definition 3.4 *A set of sentences* Π *is* **internally consistent** *under a deductive framework* $(\mathcal{L}, \vdash)$ *if it is the case that* $\Pi \subseteq \mathcal{L}$ *and there exist no two inferences* $\Pi \vdash \varphi$ *and* $\Pi \vdash \overline{\varphi}$ *where* φ *and* $\overline{\varphi}$ *are mutually exclusive (i.e.* $\Pi \vdash \neg(\varphi \wedge \overline{\varphi})$*).*

Once an agent starts relying on inferences built on assumptions however, it increases the risk that the beliefs which logically follow will *not* be internally consistent. It may not even be practical to prevent this ever happening — the inherent intractability of consistency checking for arbitrary theories described using many expressive logics means that it will not always be feasible to identify all conflicting beliefs within a knowledge base of a particular size within a given time frame, nor may it even be a simple matter to resolve them once found [Hansson, 2003]. Nonetheless, within certain limited contexts, it may still be feasible to construct a consistent theory.[5]

Definition 3.5 *A* **context theory** Π *is a set of beliefs drawn from a hypothesis space* Δ *which is internally consistent under a deductive framework* $(\mathcal{L}, \vdash)$.

The context in which a theory is formed and used is deliberately kept ill-defined here, being something determined by the reasoner and its circumstances. Fundamentally, a context theory is produced by some formal mechanism drawing from a limited hypothesis space, where the hypothesis space describes the hypotheses generated by some hypothesis generator and the conclusions which we are interested in trying to draw within the context; we then use that mechanism to select hypotheses to use as assumptions in some (hopefully) consistent theory. For example in §3.2.4, we define contexts for theories generated within an assumption-based argumentation framework.

In general, we suspect that agents in true information-rich environments will require some means to partition their beliefs so as to limit the amount of computation involved when conducting particular reasoning tasks — whatever the result of such a partitioning, the basis for that partitioning will provide the context for a given reasoning problem. In this thesis, we are interested in theories drawn within the context of practical interactions between peers:

[5] In belief revision literature, 'theory' is generally used to refer to the closure of a set of beliefs. Here it is used generically to describe any coherent set of related beliefs, which may or may not be closed under deduction.

Example 3.1 *An agent Eliza intends to enlist one of her peers to gather astronomical data for analysis in order to test her thesis. In order to determine the best course of action, she needs to produce a theory which can provide an answer to certain questions:*

> *Is the local observatory suitable for gathering the required data?*
>
> *Is there a peer who can gather the data for her?*

Eliza has already observed the following, interpreted according to some logic $\mathcal{L}$*:*[6]

> $\forall X,Y,Z.\ \mathsf{suitable}(X,Y) \wedge \mathsf{requires}(Y,Z) \rightarrow \mathsf{available}(X,Z)$
> *"If a facility is suitable for a task, then any instrument required for the task must be provided by the facility."*
>
> $\forall X,Y,Z.\ \mathsf{assignable}(X,Y) \wedge \mathsf{suitable}(Y,Z) \rightarrow \mathsf{performable}(X,Z)$
> *"An agent can perform a task if it can be assigned the task at a facility suitable for that task."*
>
> $\forall X,Y.\ \mathsf{performable}(X,Y) \rightarrow \mathsf{capable}(X,Y)$
> *"Being allowed to perform a task implies that an agent is capable of performing that task."*

Using these observations as an information source, Eliza might then generate the following hypotheses within a hypothesis generator $(\mathcal{L}, \vdash_H)$ *(such that each hypothesis is inferred from the above observations):*

> $\mathsf{suitable}(\mathsf{observatory},\mathsf{experiment})$
> *"The local observatory is suitable for Eliza's experiment."*
>
> $\mathsf{requires}(\mathsf{experiment},\mathsf{telescope}(\mathsf{optical}))$
> *"Eliza's experiment requires an optical telescope."*
>
> $\neg\mathsf{available}(\mathsf{observatory},\mathsf{telescope}(\mathsf{optical}))$
> *"The telescope at the observatory is broken."*
>
> $\mathsf{assignable}(\mathsf{dante},\mathsf{observatory})$
> *"Dante can be assigned tasks at the observatory."*
>
> $\neg\mathsf{capable}(\mathsf{dante},\mathsf{experiment})$
> *"Dante is not capable of performing this kind of experiment."*
>
> $\forall X,Y.\ \mathsf{experience}(X,Y) \rightarrow \mathsf{capable}(X,Y)$
> *"Experience implies capability."*
>
> $\mathsf{experience}(\mathsf{dante},\mathsf{experiment})$
> *"Dante has experience performing experiments like Eliza's."*

Assuming that Eliza is not able to access another information source which can confirm or refute any of these hypotheses by observation, it then befalls Eliza to produce a theory consistent with what evidence she does have which can be used to resolve the

[6] All logical expressions in this chapter use the conventions of Prolog for variables and constants, where variable names begin with an upper-case letter, and constant terms do not.

questions presented to her; this can be done by selecting a subset of the above hypotheses which is internally consistent according to a deductive *framework* $(\mathcal{L}, \vdash_D)$ *(along with any other hypotheses Eliza might generate from further observations). Our concern over the rest of the chapter then is with* how *Eliza can use* $(\mathcal{L}, \vdash_D)$ *to efficiently select which hypotheses to assume as part of a theory by which she then might use to make decisions when interacting with her peers.*

Our primary concern in this chapter is not so much with the *generation* of hypotheses, for which many approaches can be conceived, but with the rational *selection* of hypotheses consistent with available evidence and other selected hypotheses within some logical context. Given a justifiable set of assumptions, we can then form theories which can be used to describe an agent's chosen beliefs within that context, and so be used by that agent to make decisions. We are also interested in how those theories should be revised in the face of new evidence and new, possibly more compelling, propositions.

We are still concerned with one type of hypothesis generation actually. Hypotheses generated by communication, where agents engage in dialogue and introduce new concepts to one another is every bit as valid a means of hypothesis generation as private speculation, particularly because such concepts have generally already been evaluated by another peer based on its own information sources. We shall see later that the portrayal mechanism introduced in Chapter 1 can be seen as a social hypothesis generator for the agents involved in an interaction just as easily as it can be seen as a hypothesis selector for communal reasoning. It is simply a matter of perspective.

For the remainder of this chapter and the next, we shall focus on argumentation, a particular form of defeasible reasoning, which can be applied to the problem of theory generation within the context of multi-agent interaction. In particular, we demonstrate how a body of evidence can be interpreted differently based on the bias and scepticism of the observing agent, why systems of arguments can be particularly suitable for describing the relationship between hypotheses in a dynamic environment, and why argumentation is particularly applicable for managing claims made by heterogeneous agents of varying degrees of expertise in arbitrary domains provided that we can control the space in which argumentation occurs.

3.2 Using Argumentation to Rationalise Beliefs

The study of argumentation concerns itself with logical disputation — the generation of arguments in support of particular claims, and furthermore the evaluation of the

acceptability of such arguments in tandem. An argument can be considered to be a non-monotonic 'proof' for some claim insofar as it provides a basis for inferring that claim, but equally it does not preclude the possibility that that proof might be later cast into doubt. Stated differently, an argument is an appeal to the validity of a claim in the absence of further evidence to the contrary — such contrary evidence taking the form of further arguments, which may support contradictory claims or may undermine the basis upon which a previous claim is made. Such counter-arguments can themselves be attacked however, often resulting in webs of conflict which must be appraised as a whole in order to ascertain the acceptability of the assertions woven within. Acceptance however is dependent not just on the lay of arguments, but on the scepticism of the evaluator and the manner by which mutually-opposing claims are prioritised. Argumentation in informatics seeks to provide computable formalisms for every aspect of logical disputation, and so provide useful models for agent-oriented automated reasoning.

Argumentation is based on a metaphor of human discourse, and as such is often applied in the form of literal dialectic protocols for artificial multi-agent communication, particularly for negotiation [Rahwan et al., 2003] or knowledge sharing [Black and Hunter, 2007] tasks. Equally, argumentation is applied in a more purely epistemological sense as a generic paradigm for defeasible reasoning [Dung, 1995, Kowalski and Toni, 1996], wherein a reasoner essentially plays its own devil's advocate in order to ensure that the inferences drawn from its hypotheses are collectively coherent [Kakas et al., 1998, Kakas and Toni, 1999]. It can be seen however that these two approaches remain conceptually close, and from a certain abstract viewpoint merely represent a differing emphasis between the procedural and declarative levels of disputation [Prakken, 1995].

The purpose of an argumentation framework is to manage speculative (and thus defeasible) reasoning by providing a logical context in which hypothetical propositions can be tested against known facts and observations as well as against each other. It does this by producing a system of arguments specifying the relationship between individual arguments, which can then be evaluated in a number of ways. In §3.2.1 we specify argumentation in abstract, defining those properties generic to any system of arguments regardless of the underlying mechanism for generating it. In §3.2.2 we use assumption-based argumentation to define a framework which can be used to explore a particular hypothesis space, generating a system of arguments; we then in §3.2.3 specify how the argument system can be interpreted in order to derive a defeasible theory. Finally, in

§3.2.4, we summarise how all of this applies to *internal* argumentation.

3.2.1 Abstract Argumentation

Before considering different modes of argumentation, it is necessary to consider their commonalities. *Abstract argumentation* [Dung, 1995] concerns the analysis of arguments based purely on the relationships which exist between them, ignoring both the internal structure of arguments and their provenance:[7]

Definition 3.6 *A* **system of arguments** *is a pair* $(\mathcal{A}, \rightharpoonup)$ *where:*

- $\mathcal{A}$ *is a finite set of (possibly abstract) arguments.*
- $\rightharpoonup$ *is an attack relation between pairs of arguments* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$ *such that if* $\mathbf{a} \rightharpoonup \mathbf{b}$*, then "***a** *attacks* **b***".*

An argument is a statement about the world. The nature of an attack $\mathbf{a} \rightharpoonup \mathbf{b}$ is to assert that if one accepts argument **a**, then consequently one must reject argument **b**.

Example 3.2 *Consider a set of arguments exploring an extended hypothesis space based on the hypotheses generated in Example 3.1:*

a = *Dante can be assigned tasks in the observatory and the observatory can be used for this experiment, so Dante can perform the experiment (attacks* **b***).*

b = *Dante does not know how to handle this type of task, so Dante cannot perform the experiment (attacks* **a***).*

c = *The experiment requires the use of the observatory's main telescope, but the telescope is currently broken, so the observatory cannot be used for this experiment (attacks* **a***).*

d = *All instruments on the observatory are non-functional, so no agent can assign tasks to the observatory (attacks* **a** *and* **g***).*

e = *Dante has handled analogous tasks in the past, so Dante knows how to perform this type of task (attacks* **b***).*

f = *The experiment can be performed using a radio telescope, so the experiment does not require the use of an optical telescope (attacks* **c***).*

g = *Other agents are assigning tasks involving the main telescope to the observatory, so the telescope is not broken (attacks* **c** *and* **d***).*

h = *The experiment requires observations made in the visible spectrum, so the experiment cannot be performed using a radio telescope (attacks* **f***)*

This forms the system of arguments illustrated by Figure 3.4, wherein each node represents an argument and each directed edge an attack by one argument towards another.

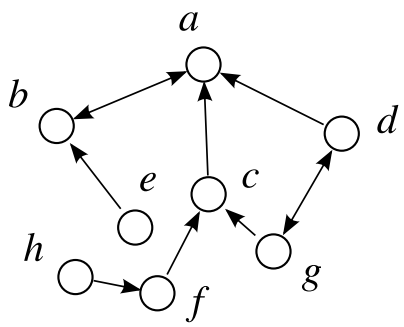


Figure 3.4: *A graph depicting the system of arguments detailed in Example 3.2.*

A key insight regarding abstract argumentation is that although arguments do not really have meaning without context, the status of arguments within an argument system can be evaluated independently of the framework in which they are embedded by considering them purely in terms of their relationship with one another. For instance:

Definition 3.7 *Given a system of arguments* $(\mathcal{A}, \rightharpoonup)$*, a set of arguments* $\mathcal{S} \subseteq \mathcal{A}$ **defends** *any argument* $\mathbf{a} \in \mathcal{A}$ *if there is an argument* $\mathbf{b} \in \mathcal{S}$ *for every attack* $\mathbf{c} \rightharpoonup \mathbf{a}$ *(where* $\mathbf{c} \in \mathcal{A}$*) such that* $\mathbf{b} \rightharpoonup \mathbf{c}$*.*

It is possible to identify particular sets of arguments which collectively describe positions which an agent might adopt in a (possibly hypothetical) debate. *Acceptability semantics* [Dung, 1995] concern themselves with the task of attributing such interpretations to abstract arguments.

Definition 3.8 *Given a system of arguments* $(\mathcal{A}, \rightharpoonup)$*,* **acceptability semantics** *confer properties onto any qualifying set of arguments, or* **extension**, $\mathcal{S} \subseteq \mathcal{A}$*. For instance:*

- $\mathcal{S}$ *is* **conflict-free** *if there are no two arguments* $\mathbf{a}, \mathbf{b} \in \mathcal{S}$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$*.*
- $\mathcal{S}$ *is* **admissible** *if* $\mathcal{S}$ *is conflict-free and* $\mathcal{S}$ *defends every argument* $\mathbf{a} \in \mathcal{S}$*.*
- $\mathcal{S}$ *is* **complete** *if* $\mathcal{S}$ *is admissible and if* $\mathcal{S}$ *defends an argument* $\mathbf{a} \in \mathcal{A}$*, then* $\mathbf{a} \in \mathcal{S}$*.*
- $\mathcal{S}$ *is* **preferred** *if* $\mathcal{S}$ *is maximally complete (under set inclusion).*
- $\mathcal{S}$ *is* **sceptically preferred** *if* $\mathcal{S}$ *is the intersection of all preferred extensions.*
- $\mathcal{S}$ *is* **grounded** *if* $\mathcal{S}$ *is minimally complete (under set inclusion).*

[7] Unless otherwise noted, all Definitions in §3.2.1 are taken from [Dung et al., 2007].

- $\mathcal{S}$ *is* **ideal** *if* $\mathcal{S}$ *is an admissible subset of the sceptically preferred extension* [Dung et al., 2007].

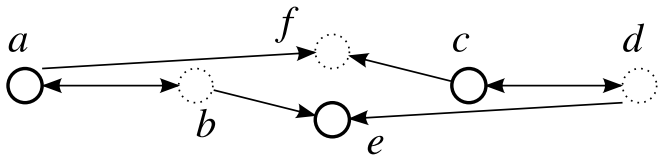
Given a particular acceptability semantic (admissibility, completeness, etc.), a set of arguments $\mathcal{S}$ *is said to be* **acceptable** *if it exhibits the chosen quality.*

Different acceptability semantics are primarily distinguished by the degree of scepticism they enforce and how comprehensively an acceptable set of arguments resolves the conflicts inherent within a given argument system. To illustrate:

- A *conflict-free* set of arguments does not attack itself, but provides no consideration for the consequences of outside arguments. It represents an untested point of view.[8]

- An *admissible* extension is an interpretation of (some of) the evidence which is defensible, but not necessarily uniquely so. It represents a point of view which is self-contained and which has not been convincingly debunked, but which may still be validly rejected in favour of some other (equally admissible) viewpoint.

  Consider the system of arguments illustrated below:

  In this instance the argument set {**a**, **c**, **e**} *is admissible, as is* {**a**}, {**c**} *and* {**a**, **c**}*, but not* {**e**}, {**a**, **e**} *or* {**c**, **e**} *which fail to defend* **e** *from both attackers* **b** *and* **d**. *Corresponding admissible sets can also be constructed from* **b**, **d** *and* **f**. {**a**, **d**} *and* {**b**, **c**} *are also admissible, as is the empty set* ∅.

  An argument is admissible if it is a member of an admissible set; if an argument is inadmissible, then it is unacceptable under any of the acceptability semantics described here other than perhaps being conflict-free.

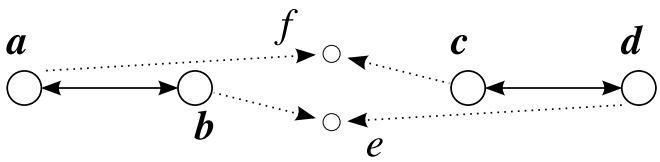
- A *complete* extension is an admissible interpretation of the evidence which includes everything in a system of arguments which it defends. It represents a confident point of view which accepts all consequences which follow from it — such a viewpoint is not inherently more correct than any other admissible interpretation however.

[8] Strictly speaking, being conflict-free is generally considered to be a prerequisite for an acceptable set of arguments rather than an actual acceptance criterion in and of itself.

From the above argument system, admissible set {**a**, **c**, **e**} *is complete, as is its mirror* {**b**, **d**, **f**}. *Sets* {}, {**a**}, {**b**}, {**c**} *and* {**d**} *are all complete, as are* {**a**, **d**} *and* {**b**, **c**}, *but* {**a**, **c**} *and* {**b**, **d**} *are incomplete without* **e** *and* **f** *respectively.*

- A *preferred* extension is a complete interpretation of as much of the evidence as can be managed whilst maintaining consistency. It represents a point of view which covers as many of the facets of a dispute as possible. If an argument system confuses more than one independent issue, then there will exist complete extensions which are not preferred extensions (because a complete extension need only fully cover the consequences of having an opinion on *some* of the issues).

  Again from the above system, {**a**, **c**, **e**} *and* {**b**, **d**, **f**} *are preferred, as are* {**a**, **d**} *and* {**b**, **c**}. *Sets* {**a**}, {**b**}, {**c**} *and* {**d**} *are* not *preferred, because more arguments can be added to each set without affecting internal consistency.*



  Note that this argument system is essentially evaluated by addressing the issues of **a** versus **b** *and* **c** versus **d**, *then looking at the resulting consequences. Every complete extension addresses one, both or neither issues, whilst every preferred extension addresses both issues by requirement.*

- The *sceptically preferred* extension is the set of claims which all coherent interpretations (at least implicitly) support. It represents the arguments acceptable from every viewpoint, but which are not necessarily self-defending (e.g. because two viewpoints have different bases upon which they accept the same argument). This extension can be controversial insofar as it accepts claims founded on uncertainty, which may be unacceptable in some circumstances (such as in legal reasoning).

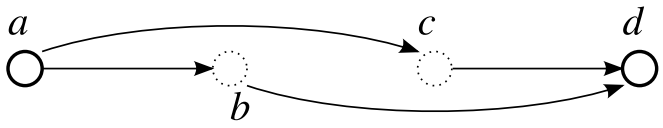
  Consider the system of arguments illustrated below:

  In this instance the argument set {**d**} *is sceptically preferred being the intersection of the preferred sets* {**a**, **d**} *and* {**b**, **d**}. *Although*

both **a** *and* **b** *defend* **d**, *they also contradict one another, meaning* **d** *is generally accepted but lacks an agreed justification.*

- The *ground* extension is the most sceptical appraisal of the evidence, accepting only arguments which are well-founded and uncontroversial. It represents a cautious point of view which accepts only those claims which appear unassailable in lieu of further arguments, and for which the justification for acceptance is universally agreed under all coherent interpretations (unlike the sceptically preferred extension).

  Consider the system of arguments illustrated below:

  In this instance the argument set $\{$**a**, **d**$\}$ *is grounded, being minimally complete —* **a** *requires no defence, so is defended by* $\{\}$*, and so must be included for completeness, but* **d** *is wholly defended by* **a**, *and so then must also be included. The resulting set is complete.*

- An *ideal* extension is a set of claims which all interpretations support, but which is also self-defending. Lying between the sceptical preferred and grounded sets in terms of scepticism, the primary difference between ground and ideal extensions (where a difference in practice even exists) lies in an ideal extension's acceptance of mutually-defending and self-defending arguments as opposed to the ground extension's insistence that every argument's defence be entirely independent of the argument itself.

  Consider the system of arguments illustrated below:

  In this instance the argument set $\{$**a**, **d**$\}$ *is ideal insofar as it is sceptically preferred and admissible. However* $\{$**a**, **d**$\}$ *is not ground because* **a** *cannot be defended without* **d**, *but* **d** *is only defended by itself or* **a**.

All measures of acceptability (alternatives of which include *stable* [Dung, 1995], *semi-stable* [Caminada, 2006b] and *eager* [Caminada, 2007] semantics) are inherently defeasible — given additional evidence, a previously accepted argument may become untenable.

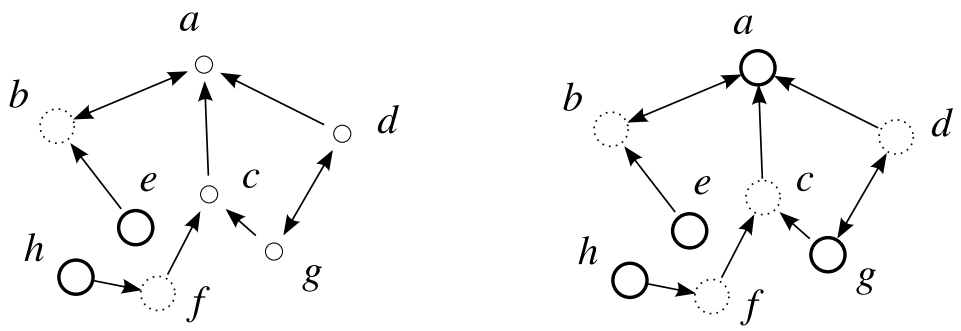


Figure 3.5: *Two interpretations of the argument system described in Example 3.2: the* ground *extension* {**e**, **h**} *attacks arguments* **b** *and* **f**, *but leaves all remaining arguments undecided; the preferred extension* {**a**, **e**, **g**, **h**} *attacks all other arguments.*

The number of possible ways a system of abstract arguments can be interpreted could be very large, since every arbitrary choice made between two opposing but admissible arguments has the potential to double the number of acceptable argument extensions which can be identified within the system. Abstract argumentation also makes no consideration of the provenance of individual arguments — in particular, the notions of acceptance described here take no account of the possibility that certain arguments may be demonstrably false (i.e. acceptance is based purely on argument relationships), or that not all attacks between arguments have been identified (i.e. argumentation is assumed to be complete within a given context). It may be that a given concrete argumentation framework will not be able to make such guarantees, in which case there may exist additional factors which will complicate the interpretation of arguments. These issues are addressed in §3.2.3.

At this point we can turn our attention back to our previous example:

Example 3.3 *There are a number of possible interpretations of Example 3.2, two of which are illustrated by Figure 3.5.*[9] *In particular, there are three complete extensions:*

- *The set* {**a**, **e**, **g**, **h**} *(illustrated on the right of Figure 3.5) is founded on the uncontroversial arguments* **e** *(which asserts that Dante knows how to perform the task in question) and* **h** *(which asserts that a radio telescope in inadequate in this case), and resolves the equally weighted dispute between* **d** *and* **g** *by favouring*

[9] An emboldened node represents an argument included in an extension, a dotted node represents an argument directly attacked by the extension. A shrunken node is one which has not been included in the extension, but which has not been formally repudiated either.

g *(accepting that the telescope is not broken). This leads to the conclusion that Dante can perform the desired experiment.*

- *The set* {**c**, **d**, **e**, **h**} *is again founded on* **e** *and* **h**, *but favours* **d** *(accepting that no agent can assign tasks to the observatory). By accepting these arguments, it cannot be said that Dante is able to perform the desired experiment.*

- *The ground extension (and thus minimal complete extension)* {**e**, **h**} *(illustrated on the left of Figure 3.5) merely accepts that Dante knows how to perform the task required and that a radio telescope cannot be used for this, but will not commit to anything else without further evidence (in particular, the question as to whether the telescope in the observatory is broken or not).*

The first two extensions above are preferred, being maximally complete. The ground extension is both sceptically preferred and ideal (which is not uncommon in practice). There are fourteen admissible extensions in total.[10]

The range of interpretations for a given argument system can be particularly broad if the system contains many mutually rebutting arguments, particularly if there are no outside arguments to decide between them. Conversely, many systems of argument in practice may have only one acceptable complete extension, which will simultaneously be the single ground, preferred, sceptically preferred and ideal extension. It is often dependent on the hypothesis space in which arguments are generated as to which tendency applies; a hypothesis space describing a domain which is highly subjective, or a space which permits one to claim the negation of any hypothesis just as easily as the hypothesis itself, is going to produce many admissible extensions and few ground arguments, whilst a more objective domain with very strongly-defined rules will produce fewer admissible arguments, and will likely have a substantial ground extension. The many different concrete argumentation frameworks produced in argumentation literature are typically justified by the perceived qualities of the domain in which they are conceived to be used.

It may be useful at this point to begin considering how an agent might derive an interpretation from a system of arguments. An *argument labelling* is a mapping of arguments to some state. Such a labelling constitutes an interpretation of an argument system, deciding whether a given argument has been accepted, rejected or left unde-

[10] Left as an exercise to the reader ...

cided:[11]

Definition 3.9 *An* **abstract argument labelling** *of a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *can be performed by a partial function* $\mathsf{lab} : \mathcal{A} \rightharpoonup \{\mathsf{in}, \mathsf{out}\}$ *where:*

- $\mathsf{lab}(\mathbf{a}) = \mathsf{in}$ *for some argument* $\mathbf{a} \in \mathcal{A}$*, if and only if for all arguments* $\mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{b} \rightharpoonup \mathbf{a}$*, it is the case that* $\mathsf{lab}(\mathbf{b}) = \mathsf{out}$.
- $\mathsf{lab}(\mathbf{a}) = \mathsf{out}$ *for some argument* $\mathbf{a} \in \mathcal{A}$*, if and only if there exists an argument* $\mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{b} \rightharpoonup \mathbf{a}$ *and* $\mathsf{lab}(\mathbf{b}) = \mathsf{in}$.

The set of arguments $\{\mathbf{a} \in \mathcal{A} \mid \mathsf{lab}(A) = \mathsf{in}\}$ *is an admissible extension of* $(\mathcal{A}, \rightharpoonup)$.

The above definition does not necessarily produce a unique labelling if any cycles exist in the argument system. As has been proven in [Caminada, 2006a], by choosing a particular labelling which maximises or minimises the assignment of different states (i.e. in, out or undecided), different types of extension arise. For example:

Any labelling — Any set of arguments labelled in according to Definition 3.9 is an admissible extension.

Any complete labelling – If a labelling of arguments is complete (such that there exist no unlabelled arguments which can be immediately labelled according to the rules of Definition 3.9 without resorting to making any more arbitrary decisions), then the set of arguments labelled in is a complete extension.

Maximal in **or** out — By attempting to label as many arguments as possible, the set of arguments labelled in will be a preferred extension.

Minimal in **or** out — By not making any arbitrary labelling decisions, the set of arguments labelled in will be the ground extension.

Intersection of maximal labellings — By labelling in every argument which is in for all preferred labellings, the set of arguments labelled in will be sceptically preferred; the maximal admissible subset of that set is ideal (and thus the sceptically preferred and ideal extensions require the most computation to label).

We can show how we might label the arguments of Example 3.2 to produce the the interpretations described in Example 3.3:

[11] Definition 3.9 is from [Caminada, 2006a], modified slightly to use a partial function into $\{\mathsf{in}, \mathsf{out}\}$ rather than total function into $\{\mathsf{in}, \mathsf{out}, \mathsf{undec}\}$.

Example 3.4 *Look again at Figure 3.5 — we have already provided two labellings of the system of arguments described by Example 3.2, using the visual convention of emboldening nodes representing arguments which are* in*, dotting out nodes for arguments which are* out*, and reducing nodes for arguments which are left undecided.*

Ground Extension — *We label* in *any arguments which are automatically acceptable (***e** *and* **h***), and then label* out *the arguments they attack (***b** *and* **f***). This may leave other arguments free of acceptable attacks (none in this case), which we can label* in*, which may lead to further arguments being labelled* out *— this can be continued until there are no more unequivocal arguments. The remainder are left unlabelled.*

Preferred Extension — *We essentially label what we can (***e** $\rightarrow$ in *so* **b** $\rightarrow$ out, **h** $\rightarrow$ in *so* **f** $\rightarrow$ out*), then one by one we make any arbitrary decisions necessary to continue labelling (label* **g** $\rightarrow$ in *so* **c**, **d** $\rightarrow$ out*) until no more arguments can be labelled* in *without contradicting Definition 3.9, or all arguments have been labelled (label* **a** $\rightarrow$ in *because* **b**, **c** *and* **d** *are* out*).*

The choice of which acceptability semantic to apply when interpreting a system of arguments is determined best by the domain from which arguments are drawn and the investment a given agent has in the acceptance or rejection of particular arguments. For example, if an agent is considering possible resolutions of constraints imposed on an interaction, and declaring the satisfaction of certain constraints carries a commitment which weighs heavily on that agent (perhaps due to difficulty in obtaining the outcome mandated by the interaction protocol, or because there is a large penalty for failing to attain the mandated outcome), then an agent may choose to interpret arguments which satisfy those constraints sceptically. Such scepticism need not be applied universally across an argument system however; one can choose arbitrarily between equally admissible arguments in one part of the system, and refuse to make a decision in another, producing a complete extension which is neither preferred (interpreting the whole system) or ground (refraining from anything controversial). Even if an agent uses an extension which is not complete however, any decisions made should be compatible with the complete extension of which that extension is a subset — the only really valid reason not to explicitly accept a proposition which is defended by an agent's accepted extension is in the case where the agent simply has not yet tried to satisfy (and thus test) that proposition.

Given an acceptability semantic, there still remains however the question of how to choose between two equally acceptable extensions (e.g. two different preferred extensions). If we assume that any purely logical reason to accept one extension over another would be expressed in the argument system itself, there must then be a heuristic element to selection which is dependent on the context in which argumentation is conducted. We shall return to this in §3.2.3, when we consider argumentation no longer in abstract.

Given then an interpretation of an argument system according to some acceptability semantic, the status of many of the propositions used within the arguments making that system can be determined. This however requires a concrete framework in which we can construct arguments from propositions so as to then later be able to deconstruct them, and so be able to construct a theory from those propositions.

3.2.2 A Framework for Constructing Arguments

The purpose of an *argumentation framework* is to provide a logical context for a system of arguments so that it can act as a vehicle for defeasible reasoning. Arguments are given a formal representation and attacks are defined in terms of how the claim made by one argument interferes with the support for another. [Prakken, 1995] identifies five elements which must be present if an argumentation framework is to be used to derive a theory from a system of arguments:

1. An *underlying formal logic* in which to express logical assertions.

2. A *notion of argument* in order to construct arguments from such logical assertions.

3. A *notion of conflict* in order to identify how one argument can attack another.

4. A *notion of defeat* in order to determine the outcome of conflicts between arguments.

5. A *notion of acceptance* in order to determine the belief status of different logical assertions given the relationships between arguments made.

In [Dung, 1995], an argumentation system is considered to consist of two components — the *argument generation* unit, and the *argument processing* unit. We will observe the same divide, considering the first three of Prakken's elements in our specification

of argument generation in this section and the final two elements in our specification of argument interpretation in §3.2.3.

Central to our particular formalisation of an argumentation framework is the notion of an *argument space*, which defines the hypothesis space which arguments exist to explore, and thus limits the detail and scope of arguments permissible within a given argumentation framework. Abstactly, an argument space can be defined as the set of all valid arguments which can be included in a given system of arguments. More practically, an argument space defines the properties which identify a valid argument; these properties may be dependent on external factors and the state of existing arguments. Thus, we can generically define an argumentation framework as follows:

Definition 3.10 *An* **argumentation framework** *can be described by a tuple* $(\mathcal{L}, \vdash, \Delta)$ *where:*[12]

- $(\mathcal{L}, \vdash)$ *is a deductive framework used to construct arguments.*
- Δ *is the* **argument space** *within which arguments are generated such that* Δ *can be treated as the set of all valid arguments.*
- *An* **argument** *is a pair* $\langle \Phi, \alpha \rangle$ *where* $\Phi \subseteq \mathcal{L}$ *forms the minimal, initernally-consistent support for a claim* $\alpha \in \mathcal{L}$ *such that* $\Phi \vdash \alpha$*, there exists no subset* $\Psi \subset \Phi$ *such that* $\Psi \vdash \alpha$ *and* $\langle \Phi, \alpha \rangle \in \Delta$*.*
- *An argument* $\langle \Phi, \alpha \rangle$ **attacks** *another argument* $\langle \Psi, \beta \rangle$ *if and only if* $\Psi \vdash \gamma$ *and* $\{\alpha\} \vdash \neg\gamma$*.*
  - *If* $\beta = \gamma$*, then* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *and* $\langle \Phi, \alpha \rangle$ **rebuts** $\langle \Psi, \beta \rangle$*.*
  - *If* $\beta \neq \gamma$*, then* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *and* $\langle \Phi, \alpha \rangle$ **undercuts** $\langle \Psi, \beta \rangle$*.*[13]

This description of an argumentation framework provides the underlying logic, notion of argument and notion of conflict necessary to generate a system of arguments which can then be evaluated in accordance with an agent's knowledge base. It should be noted that, despite being part of a greater mechanism for defeasible (non-monotonic) reasoning, within our argumentation framework we use a deductive (monotonic) framework for generating arguments in line with the reasoning articulated in [Prakken, 1995]

[12] This particular depiction of an argumentation framework is adapted from that of an *assumption-based* framework [Bondarenko et al., 1997, Dung et al., 2007]; here, any assumption set A is subsumed in the definition of argument space Δ, whilst any contraries C will be subsumed by operator $\vdash$.

[13] The notion that arguments can be either rebutted or undercut, and that this sufficiently describes all ways in which an argument can be attacked is articulated in [Pollock, 1995].

(this is standard in assumption-based argumentation [Bondarenko et al., 1997]). Essentially, the non-deductive part of argumentation is in the generation (and selection) of hypotheses with which to construct arguments, whilst the actual claims made by arguments are derived monotonically *assuming* the chosen supporting hypotheses.

Monotonicity aside, the logical framework $(\mathcal{L}, \vdash)$ used for argumentation may subsume certain axiomatic assertions beyond those rules needed to be a functional deductive system (rules like *modus ponens*) in order to simplify the argumentation process. This may include certain 'undeniable' facts and rules — 'strict' rules as distinguished from the *defeasible rules* (e.g. as used in [García and Simari, 2004]) which might be explicitly invoked within arguments (and thus can be subject to contradiction and attack):

Example 3.5 *Assume that an agent is able to make the following inference using a deductive framework* $(\mathcal{L}, \vdash)$*:*

$$\{\mathsf{observer}(\mathsf{eliza})\} \vdash \mathsf{agent}(\mathsf{eliza})$$

Assume that this is because the rule "$\forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$*" is inherently assumed by* $\vdash$ *such that* $\emptyset \vdash \forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$ *(i.e. no evidence is required to support this rule).*

If this deductive framework is used in an argumentation framework $(\mathcal{L}, \vdash, \Delta)$*, then no argument* $\langle \Phi, \alpha \rangle$ *supported by the statement "*$\forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$*" would need to explicitly include "*$\forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$*" within its support* Φ[14] *— however it also would not be possible to contradict the rule. For example, consider the following argument:*

$$\langle \Psi,\ (\mathsf{observer}(\mathsf{falstaff}) \ \wedge\ \neg \mathsf{agent}(\mathsf{falstaff})) \rangle$$

This argument is internally inconsistent because $\Psi \vdash \neg(\forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X))$ *and* $\Psi \vdash \forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$ *simultaneously; this is because we have already established that* $\emptyset \vdash \forall X.\ \mathsf{observer}(X) \rightarrow \mathsf{agent}(X)$ *and* $\vdash$ *is monotonic.*

A logical framework may also provide an expanded notion of mutual exclusion beyond simple negation:

Example 3.6 *Consider an object which may be either red, green, or blue. A naive set of arguments can be constructed to describe its possible appearance:*[15]

[14] This notion of a 'sub-minimal' argument which leaves aside established facts and rules is referred to as an 'enthymeme' [Black and Hunter, 2008].

[15] For both clarity and brevity, many example arguments in this section have been kept partially or wholly abstract — it is left to the reader to imagine how abstractions might be instantiated.

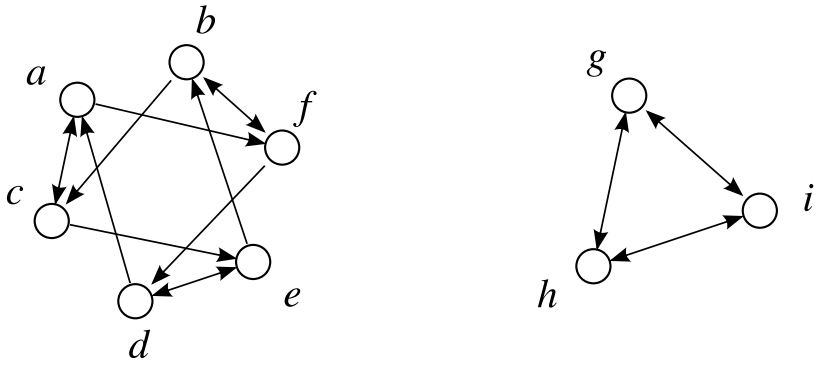


Figure 3.6: *Two systems of arguments as described in example 3.6: on the left, a naive system; on the right, the same argument system with the exclusivity of colours recognised.*

a = $\langle$ { *The object is blue,* $\forall X.\ \neg(\mathsf{blue}(X) \wedge \mathsf{red}(X))$ }, $\neg\mathsf{red}(\mathsf{thing})$ $\rangle$
b = $\langle$ { *The object is green,* $\forall X.\ \neg(\mathsf{green}(X) \wedge \mathsf{red}(X))$ }, $\neg\mathsf{red}(\mathsf{thing})$ $\rangle$
c = $\langle$ { *The object is red,* $\forall X.\ \neg(\mathsf{blue}(X) \wedge \mathsf{red}(X))$ }, $\neg\mathsf{blue}(\mathsf{thing})$ $\rangle$
d = $\langle$ { *The object is green,* $\forall X.\ \neg(\mathsf{blue}(X) \wedge \mathsf{green}(X))$ }, $\neg\mathsf{blue}(\mathsf{thing})$ $\rangle$
e = $\langle$ { *The object is blue,* $\forall X.\ \neg(\mathsf{blue}(X) \wedge \mathsf{green}(X))$ }, $\neg\mathsf{green}(\mathsf{thing})$ $\rangle$
f = $\langle$ { *The object is red,* $\forall X.\ \neg(\mathsf{green}(X) \wedge \mathsf{red}(X))$ }, $\neg\mathsf{green}(\mathsf{thing})$ $\rangle$

The resulting argument system is illustrated by Figure 3.6 on the left. Alternatively, if we factor the rule that if an object is of one colour, then it is not of another (e.g. $\emptyset \vdash \forall X.\ \neg(\mathsf{blue}(X) \wedge \mathsf{red}(X))$*) into our argumentation framework, then we can produce the system illustrated on the right of Figure 3.6:*

g = $\langle$ { *The object is red* }, $\mathsf{red}(\mathsf{thing})$ $\rangle$ *(attacks* **h** *and* **i***).*
h = $\langle$ { *The object is blue* }, $\mathsf{blue}(\mathsf{thing})$ $\rangle$ *(attacks* **g** *and* **i***).*
i = $\langle$ { *The object is green* }, $\mathsf{green}(\mathsf{thing})$ $\rangle$ *(attacks* **g** *and* **h***).*

This is much simpler and therefore easier to interpret, particularly if only part of a greater system of arguments.

Thus a logical framework which subsumes more information can be used to produce simpler systems of arguments. On the other hand, anything subsumed by the framework cannot be subject to argument if it is not shown within the argument system (i.e. subsumed sentences will not be defeasible). As we shall see later, in a dynamic system there needs to be a distinction between indelible facts (strict axioms and rules which can be safely subsumed by the framework) and things which are known, but are subject to change (defeasible rules and hypothetical assumptions which are better

left explicit in arguments and accounted for during argument interpretation). Furthermore, for multi-agent argumentation, the restriction on what we can subsume becomes even tighter — only common knowledge [Halpern and Moses, 1990] can be safely subsumed, because in practice, agents will provide their own logical frameworks and therefore might not be able to follow arguments which do not clearly show the assumptions being made.

Of course we do not want to simply generate any and every argument which can be articulated using a logical framework — we only care about arguments which have some bearing on the subject of discourse. This is ensured by the argument space of the argumentation framework, which defines the hypothesis space which arguments exist to explore. For example, in an *assumption-based* argumentation framework (as introduced in [Bondarenko et al., 1997]), an argument space is defined by the set of assumptions which can be used to support claims — any argument which draws wholly on the given assumption set (in conjunction with any strict rules subsumed by the logical framework used to infer claims from support) is therefore within the argument space, whilst any argument which uses premises not in the set is not within the space. For our purposes, we shall define a 'default' argument space as being based on a set of assumptions which can be used to support claims and a set of 'base' claims which argumentation should be focused on determining the truth of:

Definition 3.11 *The* **argument space** Δ *of an assumption-based argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *can be described by a pair* (H, F) *where:*

- *The* **horizon** $H \subseteq \mathcal{L}$ *defines the set of sentences which can be used as premises for arguments in* Δ*.*
- *The* **focus** $F \subseteq \mathcal{L}$ *defines the set of claims which* $(\mathcal{L}, \vdash, \Delta)$ *has been employed to determine the status of.*

An argument $\langle \Phi, \alpha \rangle$ *is* **within** *an argument space (i.e.* $\langle \Phi, \alpha \rangle \in \Delta$*) if and only if* $\Phi \subseteq H$ *and either* $\alpha \in F$ *or there exists an attack* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$ *such that* $\mathbf{a} \in \Delta$ *independent of* $\langle \Phi, \alpha \rangle$.

This definition of an argument space defines two things; it defines the set of hypotheses which can be used to construct arguments, and it defines the subject of argumentation. This is not the only way an argument space can be defined however. Sometimes there may be no particular focus to argumentation, or there may be other criteria by which arguments are filtered. Regardless, the point of an argument space is to ensure that any

argument made in an argumentation framework contributes to the problem it exists to solve.

Example 3.7 *Consider an argument space which subsumes the hypotheses of Example 3.1. We state that* $\Delta = (H, F)$ *where the horizon* H *is defined by the following formulae:*[16]

$\mathsf{assignable}(X_1, X_2)$

$\mathsf{suitable}(X_3, \mathsf{experiment})$

$\forall X, Y, Z.\ \mathsf{assignable}(X, Y) \wedge \mathsf{suitable}(Y, Z) \rightarrow \mathsf{performable}(X, Z)$

$\neg\mathsf{capable}(X_4, \mathsf{experiment})$

$\forall X, Y.\ \mathsf{experience}(X, Y) \rightarrow \mathsf{capable}(X, Y)$

$\mathsf{experience}(X_5, \mathsf{experiment})$

$\mathsf{requires}(\mathsf{experiment}, X_6)$

$\neg\mathsf{available}(X_7, X_8)$

$\mathsf{usable}(X_9, \mathsf{experiment})$

$\neg\mathsf{viewable}(X_{10}, X_{11})$

The focus F *of* Δ *is on any satisfaction of the following formulae, or any satisfaction of their negations:*

$\mathsf{suitable}(X_{12}, \mathsf{experiment})$

$\mathsf{performable}(X_{13}, \mathsf{experiment})$

Assume that we have a deductive framework $(\mathcal{L}, \vdash)$ *for first-order predicate logic, which has been augmented with the following domain rules:*

$\forall X, Y, Z.\ \mathsf{suitable}(X, Y) \wedge \mathsf{requires}(Y, Z) \rightarrow \mathsf{available}(X, Z)$

$\forall X, Y.\ \mathsf{assignable}(X, Y) \rightarrow \exists Z.\ \mathsf{available}(Y, Z)$

$\forall X, Y.\ \mathsf{performable}(X, Y) \rightarrow \mathsf{capable}(X, Y)$

$\forall X, Y, Z.\ \mathsf{requires}(X, Y) \wedge \mathsf{usable}(Z, X) \rightarrow Y = Z$

$\forall X, Y, Z.\ \exists A.\ \mathsf{assigns}(A, X, Y) \wedge \mathsf{requires}(Y, Z) \rightarrow \mathsf{available}(X, Z)$

$\forall X, Y, Z.\ \mathsf{requires}(X, Y) \wedge \mathsf{usable}(\mathsf{telescope}(Z), X) \rightarrow \mathsf{viewable}(\mathsf{telescope}(Z), Y)$

We can now produce concrete instances of the abstract arguments described in Example 3.2:

$$\mathbf{a} = \langle\ \{\ \mathsf{assignable}(\mathsf{dante}, \mathsf{observatory}),\ \mathsf{suitable}(\mathsf{observatory}, \mathsf{experiment}),\ \forall X, Y, Z.\ \mathsf{assignable}(X, Y) \wedge \mathsf{suitable}(Y, Z) \rightarrow \mathsf{performable}(X, Z)\ \},\ \mathsf{performable}(\mathsf{dante}, \mathsf{experiment})\ \rangle$$

[16] Variables of the form X_i are free variables, which can be treated as being existentially qualified.

$\mathbf{b} = \langle \{ \neg\mathsf{capable}(\mathsf{dante}, \mathsf{experiment}) \}, \neg\mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \rangle$

$\mathbf{c} = \langle \{ \mathsf{requires}(\mathsf{experiment}, \mathsf{telescope}(\mathsf{optical})),$
$\neg\mathsf{available}(\mathsf{observatory}, \mathsf{telescope}(\mathsf{optical})) \},$
$\neg\mathsf{suitable}(\mathsf{observatory}, \mathsf{experiment}) \rangle$
...

Note that arguments **b** *and* **c** *need not show the domain rules being applied, because those rules are subsumed within the inference mechanism used by* $(\mathcal{L}, \vdash, \Delta)$*. All three arguments shown are within* Δ*; all supporting assumptions are in* H*, the claim of argument* **a** *is in* F*, and arguments* **b** *and* **c** *attack* **a***. Conversely, the following arguments cannot be generated within* $(\mathcal{L}, \vdash, \Delta)$*:*

$\mathbf{i} = \langle \{ \mathsf{assignable}(dante, laboratory) \}, \exists X.\ \mathsf{available}(laboratory, X) \rangle$

$\mathbf{j} = \langle \{ \mathsf{confidential}(\mathsf{experiment}),$
$\neg\mathsf{private}(\mathsf{observatory}),$
$\forall X, Y.\ \mathsf{suitable}(X, Y) \wedge \mathsf{confidential}(Y) \rightarrow \mathsf{private}(X) \},$
$\neg\mathsf{suitable}(\mathsf{observatory}, \mathsf{experiment}) \rangle$

Argument **i** *can be constructed using assumptions in* H*, but is irrelevant because its claim is not is* F *nor does it attack any other argument described in Example 3.2. Argument* **j** *attacks argument* **a***, but is built from assumptions not in the hypothesis space* Δ*, and so can be considered 'outside the problem definition'. Of course, if an agent was to encounter this argument (perhaps from another agent), then it might want to expand its argument space to encompass it — we consider this possibility in §4.1.2.*

In practice, the hypothesis space in which argumentation is conducted need not be fully defined upon the outset of argumentation — it may be that an argument space is refined incrementally over the course of argumentation (though one could view this as an example of a well-defined argument space which merely imposes a more complex set of requirements on when an argument is relevant). The portrayal mechanism contributed by this thesis for instance defines an argumentation system with a minimal argument space which is then extended as the motivating interaction develops. Nonetheless, it is useful to be able to formally describe the argument space in which arguments exist at any given point in an argumentation process, in order to succinctly demonstrate various useful properties.

Given a suitable argumentation framework, it should be possible to generate a system of arguments with the properties described in §3.2.1. It should then be possible to produce an interpretation of arguments from which a defeasible theory can be derived.

3.2.3 Interpreting a System of Arguments

Given a concrete system of arguments, we should be able to derive a new theory from an interpretation of that argument system using the assumptions made in accepted arguments:

Definition 3.12 *Given a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *generated using a logical framework* $(\mathcal{L}, \vdash)$ *along with an argument labelling function* $\mathsf{lab} : \mathcal{A} \rightharpoonup \{\mathsf{in}, \mathsf{out}\}$ *and a* **theory core** $\Theta \subseteq \mathcal{L}$*, the* **accepted extension** $\mathcal{E}$ *of* $(\mathcal{A}, \rightharpoonup)$ *is the set of arguments* $\{\langle \Phi, \alpha \rangle \in \mathcal{A} \mid \mathsf{lab}(\langle \Phi, \alpha \rangle) = \mathsf{in}\}$ *in which case* $\mathcal{E}$ *defines a* **defeasible theory** $\Pi = \Theta \cup (\bigcup_{\langle \Phi, \alpha \rangle \in \mathcal{E}} \Phi)$.

In the above definition, we refer to something called the *theory core*. The theory core is a body of logical sentences which represent what is already known prior to argumentation, perhaps being the product of direct observation of the environment. Naturally, if relevant to the context in which a defeasible theory is used, it should be included alongside any selected hypotheses when making decisions. We do not just mention its existence out of a sense of completeness however. The content of the theory core can have bearing on the interpretation of arguments.

It may be that an agent is able to use observations of the environment in order to immediately ascertain the truth or falsehood of certain propositions claimed by or used in support of arguments in an argument system. As we shall see, this is particularly relevant if the observed state of the environment changes and we wish to reinterpret an argument system to reflect those changes. In such an event, we would want to be able to immediately *dismiss* any arguments which contradict observations. The theory core anchors the argumentation process by defining that which is currently 'unarguable', permitting only interpretations of arguments which appear to be consistent with it:

Definition 3.13 *Given a deductive framework* $(\mathcal{L}, \vdash)$ *and a theory core* $\Theta \subseteq \mathcal{L}$*, an argument* $\langle \Phi, \alpha \rangle$ *can be* **dismissed** *if* $\Phi \vdash \varphi$ *and* $\Theta \vdash \neg\varphi$ *for some sentence* $\varphi \in \mathcal{L}$.

If an argument is dismissed, then it is *not* included in the argument system for the purposes of abstract interpretation — so (for instance) attacks against a given argument by dismissed arguments are ignored. This has an effect on the interpretation of the argument system and therefore on the labelling of arguments. Thus we update Definition 3.9:

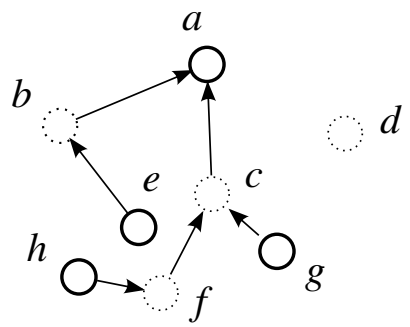


Figure 3.7: *The system of arguments from Example 3.2 after the dismissal of arguments contradicting the theory core of Example 3.7.*

Definition 3.14 *Given a theory core* $\Theta \subseteq \mathcal{L}$*, an* **argument labelling** *of a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *generated within an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *can be performed by a partial function* $\mathsf{lab} : \mathcal{A} \rightharpoonup \{\mathsf{in}, \mathsf{out}\}$ *where:*

- $\mathsf{lab}(\mathbf{a}) = \mathsf{in}$ *for some argument* $\mathbf{a} \in \mathcal{A}$*, if and only if* $\mathbf{a}$ *cannot be dismissed and for all arguments* $\mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{b} \rightharpoonup \mathbf{a}$*, it is the case that* $\mathsf{lab}(\mathbf{b}) = \mathsf{out}$.
- $\mathsf{lab}(\mathbf{a}) = \mathsf{out}$ *for some argument* $\mathbf{a} \in \mathcal{A}$*, if and only if* $\mathbf{a}$ *can be dismissed or there exists an argument* $\mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{b} \rightharpoonup \mathbf{a}$ *and* $\mathsf{lab}(\mathbf{b}) = \mathsf{in}$.

The set of arguments $\{\mathbf{a} \in \mathcal{A} \mid \mathsf{lab}(A) = \mathsf{in}\}$ *is a valid extension of* $(\mathcal{A}, \rightharpoonup)$ *given* Θ.

At this point, one might ask what the difference is between propositions in the theory core, and propositions subsumed by the logical framework with which arguments are constructed. In essence, the theory core can be changed without forcing the regeneration of an argument system, instead simply affecting an existing system's current interpretation. Thus, the theory core is better suited towards recording the environment state, which is subject to change over time, whilst we can subsume into the logical framework strict domain rules and other 'fundamentally true' propositions. Ideally, we want to subsume into the framework more knowledge so that we can simplify the system of arguments then generated. In practice however, we may find there is little that we can accept unconditionally and more which is subject to change, and so we place these things into the theory core.

Example 3.8 *Referring back to Examples 3.2 and 3.7, let us look at possible interpretations of our argument system. We have already identified the different naive interpretations of the system in Example 3.3; let us assume now that Eliza can glean additional information about her current environment:*

$\mathsf{available}(\mathsf{observatory}, \mathsf{seismograph})$
"The observatory's seismograph is operational".

$\exists X, Y.\ \mathsf{assigns}(X, \mathsf{observatory}, Y) \wedge \mathsf{requires}(Y, \mathsf{telescope}(\mathsf{optical}))$
"Someone has been assigned a task at the observatory which requires the optical telescope".

$\mathsf{requires}(\mathsf{experiment}, \mathsf{images}(\mathsf{visible}))$
"Eliza's experiment requires images in the visible spectrum."

This becomes part of the theory core for any theory derived from the argument system. We can now dismiss argument **d***:*[17]

$$\mathbf{d} = \langle\ \{\ \neg\exists X.\ \mathsf{available}(\mathsf{observatory}, X),\ \forall X, Y.\ \mathsf{assignable}(X, Y) \rightarrow \exists Z.\ \mathsf{available}(Y, Z)\ \},\ \forall X.\ \neg\mathsf{assignable}(X, \mathsf{observatory})\ \rangle$$

Argument **d** *is predicated on the assumption that all instruments in the observatory are non-functional, allowing for the claim that no agent can be assigned to it. However it has been observed that for at least one instrument, this is not the case, and thus the argument has been undercut by the theory core. The dismissal of argument* **d** *leads to the effective argument system illustrated in Figure 3.7, and leaves us with only one of our original three extensions remaining;* $\{\mathbf{a}, \mathbf{e}, \mathbf{g}, \mathbf{h}\}$*, which is now both preferred and grounded (amongst other things).*

Note that we do not directly accept arguments based on the theory core. Merely knowing that part of the support for an argument is necessarily true does not make the whole argument sound. Moreover, even if the claim of an argument is observed to be in, that does not mean the argument itself is not drawing that claim for invalid reasons — it merely means that any argument which rebuts that claim is definitely unacceptable. It is true that if the *entire* support of an argument is necessarily true, then the argument must be accepted — however we still do not need to take any additional action, because we know that every attack against that argument will be dismissed.

The interpretation of arguments allows an agent to infer something about the conflicts between possible assumptions drawn from a set hypotheses. The extent to which arguments illustrate those conflicts is dependent on the thoroughness of argumentation — the more exhaustive a system of arguments, the better it describes the argument space within which it exists. Conversely, if a system of arguments fails to explore the argument space well enough, then the conclusions which would be drawn from an abstract evaluation of the system may conceal inconsistencies:

[17] We include the subsumed rule which argument **d** relies on in the support of **d** for clarity.

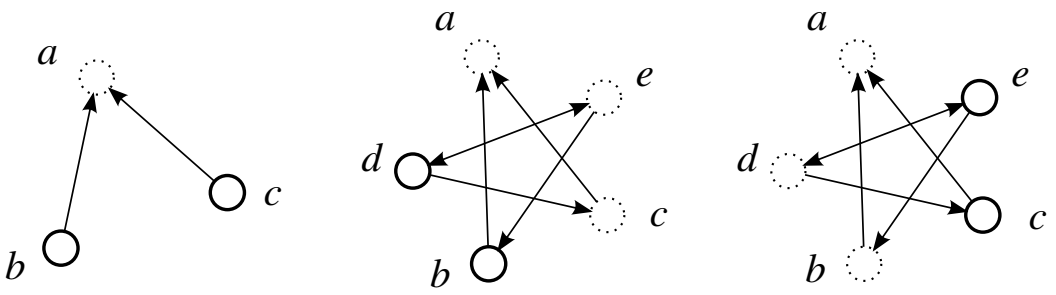


Figure 3.8: *Additional arguments can 'fix' insufficiently expressive argument systems. On the left, arguments* **b** *and* **c** *are accepted, despite being mutually exclusive. By adding two additional arguments, only one of the two arguments is acceptable (centre and right).*

Example 3.9 *Consider the following system of arguments:*

a = $\langle$ { ¬deformable, ¬fragmentable }, solid $\rangle$.

b = $\langle$ { soft, strong }, deformable $\rangle$ *(attacks* **a***).*

c = $\langle$ { sharp, brittle }, fragmentable $\rangle$ *(attacks* **a***).*

A naive interpretation of the above arguments would conclude that both arguments **b** *and* **c** *are acceptable, rejecting argument* **a**. *However if this system of arguments existed within an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *in which* $\{\mathsf{strong}, \mathsf{sharp}\} \vdash \mathsf{durable}$ *and* $\{\mathsf{soft}, \mathsf{brittle}\} \vdash \neg\mathsf{durable}$*, then it becomes clear that we cannot accept both* **b** *and* **c** *as part of the same extension — we would be able to infer both* durable *and* ¬durable *from the resulting theory.*

There are a number of different arguments which we can add in order to resolve this matter. For example, we can add two more arguments:

d = $\langle$ { soft, strong }, $\neg$(sharp $\wedge$ brittle) $\rangle$ *(attacks* **c** *and* **e***).*

e = $\langle$ { sharp, brittle }, $\neg$(soft $\wedge$ strong) $\rangle$ *(attacks* **b** *and* **d***).*

We can then resolve the inconsistency, either by accepting argument **d** *and so defeating* **c** *and* **e***, or by accepting argument* **e** *and so defeating* **b** *and* **d** *(illustrated by Figure 3.8).*

However whilst it may be that more arguments may make a system of arguments more descriptive of a given hypothesis space, it may not always be feasible to exhaustively generate a 'complete' argument system, nor may it in some contexts be even possible to be certain that a given system of arguments *is* complete. This does not invalidate the use of argumentation however. It merely means that we cannot always rely purely

on the abstract relationships between arguments when evaluating real systems of arguments.

Consider the problem of deriving consistent belief sets from a hypothesis space from another perspective — truth maintenance [Doyle, 1979]. In essence, we have a constraint satisfaction problem wherein we must determine whether or not given assertions are 'in' or 'out' in such a manner as to be totally consistent, particularly with the assertions which from the beginning we *know* are true or false. Argumentation as described here can be seen in one of two ways; as either a particular form of truth maintenance, or as a goal-directed way of simplifying the underlying truth maintenance problem. The first viewpoint requires argumentation to be 'complete', such that it fully explores a given hypothesis space, and so the interpretation of arguments after accounting for observations results in consistent theories. The second viewpoint acknowledges that systems of arguments may well be incomplete at times, and thus the theory derived from the interpretation of arguments may not actually be fully consistent.

If we take the second viewpoint, then we have to consider how we deal with a possibly inconsistent theory: we can somehow ensure completeness by completing the system of arguments (which requires exhaustive argument generation since we do not necessarily know where any unexplored conflicts actually are); we can treat the labelling of arguments as a kind of partial solution to a traditional truth maintenance problem and proceed with that; or we can accept the possibly inconsistent theory and only generate new arguments upon the identification of an inconsistency, risking that there may be earlier decisions made using the theory which were not in fact quite as rational as we had assumed. In reality, any of these approaches can be acceptable, based on the qualities of the environment in which arguments are made.

Probably the simplest way to ensure consistent theories is to allow the dismissal of additional arguments *during* argument interpretation, rather than just immediately beforehand. Basically, as we instantiate a theory based on the labelling of arguments, we continue to dismiss any arguments which contradict the partial theory:

Definition 3.15 *Given a deductive framework* $(\mathcal{L}, \vdash)$ *and a theory core* Θ *producing a theory* Π *from a system of arguments* $(\mathcal{A}, \rightharpoonup)$*, an argument* $\langle \Phi, \alpha \rangle \in \mathcal{A}$ *can be* **dismissed** *if and only if* $\Phi \vdash \varphi$ *and* $\Pi \vdash \neg\varphi$ *for some sentence* $\varphi \in \mathcal{L}$.

This declarative refinement of Definition 3.13 subsumes that prior definition, since from Definition 3.12 we know that the theory core is part of the resultant theory. Note

that there is no need to update Definition 3.14.

Example 3.10 *We have already demonstrated in Example 3.9 that adding new arguments would fix the inherent inconsistency in the interpretation of the resulting argument system. It should be clear that an interpretation of the extended argument system would produce a valid theory* Π *as per Definition 3.12.*

We can also fix the inconsistency by dismissing one of arguments **b** *and* **c** *after partially labelling the system — for instance, by accepting argument* **b***, we note that* $\{\mathsf{soft}, \mathsf{strong}\} \subseteq \Pi$ *and* $\{\mathsf{soft}, \mathsf{strong}\} \vdash \neg(\mathsf{sharp} \wedge \mathsf{brittle})$ *according to the argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *of Example 3.9. Therefore, by Definition 3.15, we see that argument* **c** *should be dismissed, because* $\{\mathsf{sharp}, \mathsf{brittle}\} \vdash (\mathsf{sharp} \wedge \mathsf{brittle})$.

There is one final point to made regarding the interpretation of arguments, one which particularly illustrates the advantages of using a system of arguments to describe the provenance of a theory. It is this — if we treat different states of the environment in much the same fashion as we treat different interpretations of a single environment state, and then rely on a (changing) body of observations to dismiss and indirectly force the acceptance of particular arguments in the resulting argument system, then we can construct a stable description of a volatile environment which can be used to maintain a coherent theory which adapts as circumstances change.

First, consider how a dynamic environment might render a particular interpretation of the environment unacceptable. As established in the previous section by Definition 3.15, an argument can be dismissed if it is already evident from already accepted propositions that the argument cannot hold. Intuitively (for internal argumentation at least), one would not expect an agent to deliberately compose an argument which it can already deduce to be false, in which case the most likely reason for the dismissal of an argument is either an unexplored conflict between two propositions or a change in the theory core. In question is what to *do* with dismissed arguments. If they are permanently removed from the system of arguments, then what happens if the environment returns to a state wherein they are again valid? It would seem best to minimise the number of revisions which a system of arguments is subjected to, in which case it may be best to retain arguments regardless of their acceptability given the current theory core on the basis that that as long as an agent can determine which arguments (and attacks) should be dismissed at any one given point, any interpretations which apply to alternate environment states will ‘collapse’ by virtue of being indefensible.

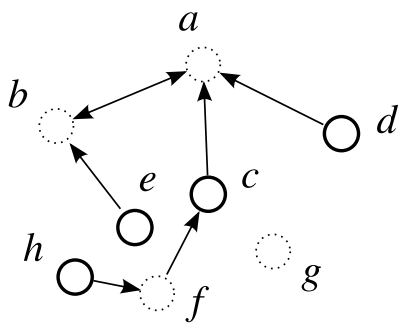


Figure 3.9: *The system of arguments from Example 3.2 after the dismissal of different arguments contradicting the theory core of Example 3.11.*

Example 3.11 *Consider yet again the system of arguments produced by the argumentation framework described in Example 3.7. In Example 3.8, we were able to dismiss argument* **d** *because it contradicted agent Eliza's theory core. Assume instead that Eliza's theory core was merely as follows:*

$\neg\exists X.\ \mathsf{available}(\mathsf{observatory}, X)$
"All of the observatory's instruments are non-functional".

In this case, we would be forced to dismiss argument **g***:*

$\mathbf{g} = \langle\ \{\ \exists X, Y.\ \mathsf{assigns}(X, \mathsf{observatory}, Y) \wedge \mathsf{requires}(Y, \mathsf{telescope}(\mathsf{optical}))\ \},$
$\mathsf{available}(\mathsf{observatory}, \mathsf{telescope}(\mathsf{optical}))\ \rangle$

This would lead to the only valid extension being $\{\mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{h}\}$*, as illustrated by 3.9. If however, at a later point the instruments were to be repaired, then the system would revert to a state in which* $\{\mathbf{a}, \mathbf{e}, \mathbf{g}, \mathbf{h}\}$ *was the only valid extension. Thus the system of arguments can describe both scenarios without modification.*

This illustrates a desirable quality of an abstract system of arguments; it can be composed in such a way that it provides a stable description of a dynamic system, with observation (i.e. the theory core) being then used to select the correct interpretation given the current environment state.

It should be noted that if an agent is unable to determine which arguments do not apply to the current environment, then interpretations of alternate states will remain admissible. This does not concern us however, because if an agent lacks the knowledge required to dismiss key arguments, then it must also lack the knowledge required to distinguish between alternative states anyway, in which case the issue becomes one of insufficient knowledge, rather than some flaw in the expressivity of the argument system.

Example 3.12 *Continuing on from Example 3.8, let us produce a defeasible theory from the only remaining valid extension of our argument system. The labelling of the system is as follows:*

in — *Arguments* **a**, **e**, **g** *and* **h**.

out — *Arguments* **b**, **c**, **d** *and* **f**.

The resulting theory is therefore the set of all assumptions used in arguments **a**, **e**, **g** *and* **h**, *along with the contents of Eliza's theory core:*

$\mathsf{assignable}(\mathsf{dante}, \mathsf{observatory})$

$\mathsf{suitable}(\mathsf{observatory}, \mathsf{experiment})$

$\forall X, Y, Z.\ \mathsf{assignable}(X, Y) \wedge \mathsf{suitable}(Y, Z) \rightarrow \mathsf{performable}(X, Z)$

$\mathsf{experience}(\mathsf{dante}, \mathsf{experiment})$

$\forall X, Y.\ \mathsf{experience}(X, Y) \rightarrow \mathsf{capable}(X, Y)$

$\exists X, Y.\ \mathsf{assigns}(X, \mathsf{observatory}, Y) \wedge \mathsf{requires}(Y, \mathsf{observatory})$

$\mathsf{requires}(\mathsf{experiment}, \mathsf{telescope}(\mathsf{optical}))$

$\mathsf{requires}(\mathsf{experiment}, \mathsf{images}(\mathsf{visible}))$

$\neg\mathsf{viewable}(\mathsf{telescope}(\mathsf{radio}), \mathsf{images}(\mathsf{visible}))$

Along with the rules subsumed within $(\mathcal{L}, \vdash, \Delta)$ *(the argumentation framework in which our arguments were generated),*[18] *we can infer the following conclusions:*

$\mathsf{performable}(\mathsf{dante}, \mathsf{experiment})$

$\mathsf{capable}(\mathsf{dante}, \mathsf{experiment})$

$\mathsf{available}(\mathsf{observatory}, \mathsf{telescope}(\mathsf{optical}))$

$\neg\mathsf{usable}(\mathsf{telescope}(\mathsf{radio}), \mathsf{experiment})$

Thus, Eliza is likely to ask Dante if he could provide the experimental data she requires from the local observatory.

Even under the influence of the theory core, the choice of semantics for determining acceptance of arguments is dependent on the needs of the agent, as discussed back in §3.2.1. It is possible to define preference orderings on concrete arguments based on their content however, making the choice between two equal interpretations of an argument system easier [Prakken and Sartor, 1995, Kowalski and Toni, 1996] — for example based on historical probability, legal precedence or even just how large an argument is. In multi-agent contexts, an agent can also make preferences based on the

[18] These rules can be explicitly added to the theory if it is to be used with a different logical framework.

provenance of assertions (one peer's word may be trusted over another) or on notions of social welfare [Rahwan and Larson, 2008].

We define the requirements for interpreting concrete systems of arguments here, but not algorithms that implement these requirements. Whilst important, our primary concern here is defining a logical model for argumentation which allows flexibility of state and hypothesis space (explored in the next section), such that we can then distribute that model under various conditions and demonstrate its theoretical feasibility as a model for our contribution. The actual process of interpreting arguments is left to the implementation of individual agents – there have been various attempts to find good algorithms for evaluating assumptions used for argumentation as well as attempts to determine their potential tractability (see [Dimopoulos et al., 1999, Vreeswijk and Prakken, 2000, Dung et al., 2007, Dunne, 2008]), albeit often subject to various limitations.

3.2.4 Defeasible Reasoning as Internal Argumentation

The process of defeasible reasoning on the part of an intelligent agent can be seen as a two part process of hypothesis generation and hypothesis selection. From a combination of introspection and observation of the environment, an agent can produce hypotheses which seek to explain phenomena found in the world. By testing these hypotheses against one another and against what is already known, an agent can determine which subsets of hypotheses collectively interpret its world in a rational, internally consistent way.

Assumption-based argumentation can be used as a hypothesis selector, generating arguments to test hypotheses and evaluating the results. The hypotheses generated by a hypothesis generator define the hypothesis space to be explored; the argument space of an argumentation framework then specifies how arguments should be used to explore that hypothesis space. The argumentation framework uses deductive logic and the rules of the domain in which the hypotheses are generated to produce arguments which fit within the given argument space — these arguments assume hypotheses within the hypothesis space in order to derive claims which can then be contrasted against those of other arguments and against any direct observations of the environment made by the agent. The agent can then interpret the system of arguments generated by applying acceptance criteria based on the conclusions it is most interested in, the scepticism by which it wishes to regard conclusions and the preference bias it extends to otherwise

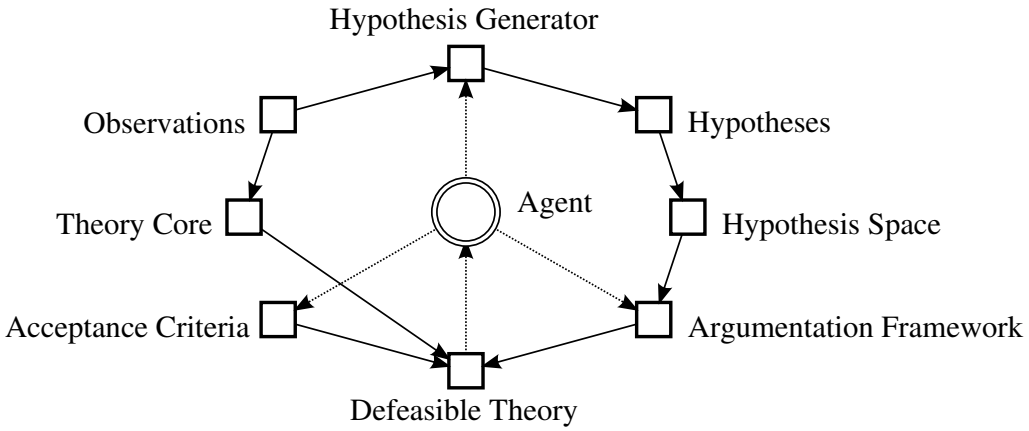


Figure 3.10: *An agent generates a theory based on observations and hypotheses interpreted using an argumentation framework.*

equal contrasting viewpoints.

In this respect, the argumentation process just described follows the methodology advocated by Imre Lakatos' philosophy of science [Lakatos, 1970], wherein a scientific programme is composed of a *hard core* and a *protective belt*. The hard core describes the fundamental characteristics of the programme, which are not in question, whilst the protective belt consists primarily of auxiliary assumptions. The key, if one is to maintain the programme, is to resolve the inconsistencies within the programme by adapting the protective belt in order to preserve the integrity of the core. In this instance the hard core is the theory core, comprised of observations of the environment and other 'unarguable' propositions (which may include technically defeasible elements which the agent has elected not to subject to argumentation) whilst the protective belt is the set of assumptions accepted as part of the interpretation of the argument system produced.

We can now pull together all the necessary components of an assumption-based argumentation system, and define the context in which an agent can produce a defeasible theory via internal argumentation. As alluded to in §3.1, the *context* of a theory describes the provenance of the propositions within that theory. From an interpretation of a system of arguments, we can infer a theory based on the assumptions used in the construction of accepted arguments. Thus, an argumentation framework, along with a record of arguments generated and a theory core, can act as context for that theory:

Definition 3.16 *The* **theory context** $\mathcal{C}$ *for a theory* Π *can be described by a tuple* $(\Theta, (\mathcal{L}, \vdash, \Delta), (\mathcal{A}, \rightharpoonup), \mathsf{lab})$ *where:*

- Θ *is the theory core of* Π.
- $(\mathcal{L}, \vdash, \Delta)$ *is an assumption-based argumentation framework in which* $(\mathcal{A}, \rightharpoonup)$ *is generated.*
- $(\mathcal{A}, \rightharpoonup)$ *is a system of arguments, the interpretation of which determines* Π*; because* $\mathcal{C}$ *is considered to be a persistent device,* $(\mathcal{A}, \rightharpoonup)$ *is stored alongside the* $(\mathcal{L}, \vdash, \Delta)$.
- *An argument labelling function* $\mathsf{lab}(\Theta) : \mathcal{A} \rightharpoonup \{\mathsf{in}, \mathsf{out}\}$ *is a partial function which interprets* $(\mathcal{A}, \rightharpoonup)$ *according to chosen acceptability semantics.*

The **accepted extension** $\mathcal{E}_A$ *of* $\mathcal{C}$ *is the set of arguments* $\{\langle \Phi, \alpha \rangle \in \mathcal{A} \mid \mathsf{lab}(\Theta, \langle \Phi, \alpha \rangle) = \mathsf{in}\}$ *in which case* $\Pi = \Theta \cup (\bigcup_{\langle \Phi, \alpha \rangle \in \mathcal{E}_A} \Phi)$.

- *The* **rejected set** $\mathcal{R}$ *of* $\mathcal{C}$ *is the set of arguments* $\{\langle \Phi, \alpha \rangle \in \mathcal{A} \mid \mathsf{lab}(\Theta, \langle \Phi, \alpha \rangle) = \mathsf{out}\}$.
- *The* **unrejected set** $\mathcal{U}$ *of* $\mathcal{C}$ *is the set of arguments* $\{\langle \Phi, \alpha \rangle \in \mathcal{A} \mid \mathsf{lab}(\Theta, \langle \Phi, \alpha \rangle) \neq \mathsf{out}\}$.

For notational convenience, if an argument $\langle \Phi, \alpha \rangle \in \mathcal{A}$ for $\mathcal{A}$ of $\mathcal{C}$, then we can simply write $\langle \Phi, \alpha \rangle \in \mathcal{C}$. Also, if $\Phi \vdash \alpha$ according to $(\mathcal{L}, \vdash)$ of $\mathcal{C}$, then $\Phi \vdash \alpha$ according to $\mathcal{C}$.

Whilst an agent can impose any criteria it wishes on argument acceptance, including acceptability semantics not specified in §3.2.1, we can reasonably assume that any accepted extension of $\mathcal{C}$ is *complete* (i.e. has no internal conflicts and contains all argument which it defends from attacks in $(\mathcal{A}, \rightharpoonup)$ as per Definition 3.7). Otherwise, if we refer to an acceptable extension $\mathcal{E}$ of $\mathcal{C}$, then $\mathcal{E}$ can be *any* admissible extension of the argument system $(\mathcal{A}, \rightharpoonup)$ in $\mathcal{C}$ which accounts for dismissible arguments as per Definition 3.15. For convenience, we also define the sets of rejected and unrejected arguments — in Chapter 5, these sets will prove useful for determining how aggressively an agent should respond to its peers' arguments within an interaction portrayal.

The notion of a theory's context is very useful. It brings together argument generation and argument interpretation, allowing us to simply refer to the context $\mathcal{C}$ of a given argumentation process. This gives us a single entity to refer to when we want to change the 'programme' of a defeasible theory. In particular, the theory context describes the source of an agent's contribution to a *social* (i.e. multi-agent) argumentation process, and the medium through which it will evaluate any shared system of arguments. Thus, we have everything we need to move on to a discussion of distributed argumentation;

an understanding of abstract argumentation, an understanding of argumentation frameworks in general, and the ability to refer to different contexts of argumentation.

The next chapter then concerns itself with the mapping of arguments from one context into another; by this means can we consider a social argumentation process to be a bridge between the private defeasible reasoning systems of individual agents. This will provide a formal basis by which to understand the portrayal of interactions described in Chapters 1 and 2, as an interaction portrayal can simply be seen as an intermediary between the personal beliefs of peers in dialogue.

Chapter 4

Distributed Multi-Agent Argumentation

Intelligence cannot exist in a vacuum. It is only by observing the world that new insights can be attained, and it is only by testing hypotheses in the world that an agent can construct a robust model of its environment. It has already been demonstrated however that it is difficult for a lone agent, with only limited access to information within its environment, to establish the true nature of its world. It would certainly be wasteful to insist that that agent try to independently determine that nature by itself when there exist other agents with which it can collaborate, giving the agent access to the observations and insights of its peers.

Argumentation is an inherently social metaphor, wherein a proponent of some claim is pitted against an aggressive devil's advocate who seeks to pull apart any loose threads found in a given argument. Yet for the most part we have treated argumentation as a process of self-correction on the part of a single agent. In this chapter, we shall redress the balance and consider argumentation as a social process, where multiple autonomous agents test their theories against those of their peers. In doing so, we shall identify many of the challenges faced in trying to conduct a coherent argumentation process between heterogeneous agents, and we shall also consider the relationship between an external system of arguments articulated by peers as part of social argumentation and the internal argument systems which provide context for the beliefs of individual agents. In doing this, we shall be able to define more formally the decision problem which our portrayal mechanism of Chapter 5 exists to solve.

4.1 Mapping Arguments Between Different Contexts

The most obvious way to regard social argumentation is as a hypothesis selection mechanism like that described in §3.2.4, albeit one in which there are multiple processes contributing to the generation and interpretation of arguments. From this perspective, we should be able to infer a distributed context in which argumentation is conducted. This distributed context would have the same components as any other theory context:

- A theory core Θ representing the unarguable within the hypothesis space of the social argumentation process.
- An argumentation framework $(\mathcal{L}, \vdash, \Delta)$ within which arguments are generated.
- A system of arguments $(\mathcal{A}, \rightharpoonup)$ contributed by the agents involved in argumentation.
- A labelling function lab by which $(\mathcal{A}, \rightharpoonup)$ can be interpreted in order to produce some common theory Π.

The flaw in this approach however is that it conflates a group of autonomous agents together into a single entity, one which processes the information given and presents a single theory. Whilst this may be fine for analysing a social argumentation process from the outside, it tells us little about social argumentation from the perspective of the peers engaged in it. How is the theory core determined? How can agents collectively decide upon a common argument framework? Can a group of autonomous peers agree on a single interpretation of a shared argument system?

Instead of viewing multi-agent argumentation as a communal hypothesis selection process, we can view multi-agent argumentation as a hypothesis *generation* process for the individual agents engaged in argumentation. The insight here is that social argumentation is merely another means by which new observations and hypotheses can be introduced to an agent. The arguments articulated by peers serve to introduce new hypotheses for the recipient to factor into its programme for deriving its theory for some problem domain — this may or may not lead to a revision of the recipient's beliefs. Those beliefs determine whether or not a counter-argument can be made, and any counter-argument might serve to introduce new hypotheses to other peers.

Of course, we have already conceived an agent's defeasible reasoning process as an argumentation process in and of itself, and this allows us for the most part to bypass

the actual theory produced by internal argumentation and instead directly compare the system of arguments generated socially with the system generated *internally* as part of the agent's theory context.

4.1.1 Mapping Arguments Across Argument Spaces

Consider the relationship between the argument space of a social argumentation process and the argument spaces in which individual agents generate their private theories. Evidently the arguments articulated by an agent must be based on arguments it can conceive privately. It is also true however that any other arguments generated by its peers must also at least *potentially* be arguments it has considered, or else it cannot be said for certain that the agent's theory truly accounts for the attacks of peers. Thus the argument space of a shared system of arguments should exist within an intersection of the argument spaces of all involved agents' theory contexts.

Of course *in theory*, if the argument space of a social argumentation process was subsumed by the theory contexts of every peer, then there would not actually be any reason to discuss anything, because each agent would already have the capability to generate any argument which could be made by its peers independently. Of course this ignores that an agent might not have fully explored the hypothesis space from which it derives its theory (recall the discussion of argumentation completeness in §3.2.3), which social argumentation could rectify by providing assistance identifying conflicts. It also ignores that agents may be able to provide additional observations of the environment, permitting the dismissal of additional interpretations of the arguments in that shared argument space (likewise discussed in §3.2.3). Finally, it ignores that fact that the agents engaging in social argumentation do not usually know precisely what the argument space for a shared argumentation process is going to be, let alone if it will lie within the intersection of their own individual theory contexts.

Potential argumentation concerns itself with the comparison of argument spaces, and the mapping of arguments from one space to another. In essence, its concern is with addressing the question of how best to articulate the same argument in different circumstances — circumstances in which the fundamental premises upon which claims can be supported are subject to different levels of abstraction, and in which factors relevant in one space are considered irrelevant in another. Our interest in this case lies in the migration of arguments from an agent's theory context into the shared argument space of a social argumentation process, and *vice versa*. Critical to this is the notion of

a *potential argument*.

A potential argument is an argument which is supported by premises which can be derived from other, more fundamental propositions, and as such could be equally well replaced by any of a number of more concrete arguments:[1]

Definition 4.1 *Given a logical framework* $(\mathcal{L}, \vdash)$*, an argument* $\langle \Phi, \alpha \rangle$ *is considered to be a* **potential argument** *in relation to another argument* $\langle \Psi, \alpha \rangle$ *if, for every sentence* $\varphi \in \Phi$*, it is the case that* $\Psi \vdash \varphi$*.*

- *For brevity, we often state this relationship as* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi, \alpha \rangle$*.*

If there exists a sentence $\varphi \in \Psi$ *such that* $\Phi \nvdash \varphi$*, then* $\langle \Phi, \alpha \rangle$ *is a* **strictly** *potential argument for* $\langle \Psi, \alpha \rangle$*.*

- *We state this as* $\langle \Phi, \alpha \rangle \sqsubset \langle \Psi, \alpha \rangle$*.*

A potential argument **a** for another argument **b** is said to be *potentially* **b**. Conversely, **b** is an *elaboration* upon **a**. This notion of a potential argument allows us to express the relationship between arguments of varying levels of detail, as well as allowing us to identify the common abstraction of two distinct arguments with matching claims. A potential argument is similar to an enthymeme [Black and Hunter, 2008] — however instead of concealing common knowledge, in a potential argument the provenance of certain propositions in the support of the argument is left to be assumed by the observer.

Example 4.1 *Consider the following collection of arguments:*

$$\mathbf{a} = \langle \{ \mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \}, \mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \rangle$$

$$\mathbf{b} = \langle \{ \mathsf{assignable}(\mathsf{dante}, \mathsf{observatory}), \mathsf{suitable}(\mathsf{observatory}, \mathsf{experiment}), \forall X, Y, Z.\ \mathsf{assignable}(X, Y) \wedge \mathsf{suitable}(Y, Z) \rightarrow \mathsf{performable}(X, Z) \}, \mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \rangle$$

$$\mathbf{c} = \langle \{ \mathsf{assignable}(\mathsf{dante}, \mathsf{observatory}), \mathsf{requires}(\mathsf{experiment}, \mathsf{telescope}(\mathsf{optical})), \mathsf{available}(\mathsf{observatory}, \mathsf{telescope}(\mathsf{optical})), \forall X, Y, Z.\ \mathsf{requires}(X, Y) \wedge \mathsf{available}(Z, Y) \rightarrow \mathsf{suitable}(Z, X), \forall X, Y, Z.\ \mathsf{assignable}(X, Y) \wedge \mathsf{suitable}(Y, Z) \rightarrow \mathsf{performable}(X, Z) \}, \mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \rangle$$

$$\mathbf{d} = \langle \{ \forall X.\ \mathsf{performable}(X, \mathsf{experiment}) \}, \mathsf{performable}(\mathsf{dante}, \mathsf{experiment}) \rangle$$

[1] This notion of potential argument is inspired by that used in [Gaertner and Toni, 2008]; the principle difference here however is that we focus on a relative relation between arguments rather than on partial constructions of arguments as part of a dispute derivation.

Assuming a deductive framework capable of interpreting the above arguments, we can deduce that:

- $\mathbf{a} \sqsubset \mathbf{b}$ *(i.e. argument* **a** *is potentially argument* **b** *and argument* **b** *elaborates upon argument* **a***).*
- $\mathbf{a} \sqsubseteq \mathbf{a}$ *(demonstrating reflexivity in the non-strict case).*
- $\mathbf{b} \sqsubset \mathbf{c}$ *and* $\mathbf{a} \sqsubset \mathbf{c}$ *(demonstrating transitivity).*
- $\mathbf{b} \not\sqsubseteq \mathbf{a}$ *and* $\mathbf{c} \not\sqsubseteq \mathbf{b}$ *(demonstrating anti-symmetry).*
- $\mathbf{a} \sqsubset \mathbf{d}$ *(although argument* **d** *is supported by a more generic assertion than for argument* **a***, argument* **d** *makes a greater commitment as to how its claim is derived than argument* **a***, and thus can be attacked by arguments which have no bearing on* **a***).*
- $\mathbf{b} \not\sqsubseteq \mathbf{d}$ *and* $\mathbf{d} \not\sqsubseteq \mathbf{b}$ *(arguments* **b** *and* **d** *are separate, distinct elaborations of* **a***; the same applies to arguments* **c** *and* **d***).*

Defining potential arguments (and inversely, elaborations) serves as a useful means to relate arguments formulated under the same logical framework, but within different argument spaces. For example, we can take an argument too detailed to fit within a given argument space, and find an abstraction which does, or we can flesh out an overly-simple argument if we are able to elaborate upon the assumptions made within it. Neither elaborations of a potential argument or potential arguments for an elaboration are necessarily unique however:

Example 4.2 *Assume an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *in which the following rules are subsumed by* $(\mathcal{L}, \vdash)$*:*

$$\forall X, Y.\ \mathsf{registered}(X,Y) \wedge \mathsf{assignable}(X,Y)$$
$$\forall X, Y, Z.\ \mathsf{requires}(X,Y) \wedge \mathsf{available}(Z,Y) \rightarrow \mathsf{suitable}(Z,X)$$
$$\forall X, Y, Z.\ \mathsf{assignable}(X,Y) \wedge \mathsf{suitable}(Y,Z) \rightarrow \mathsf{performable}(X,Z)$$

Given the following argument:

$$\begin{aligned}\mathbf{a} = \langle\ \{\ & \mathsf{registered}(\mathsf{dante}, \mathsf{observatory}),\\ & \mathsf{requires}(\mathsf{experiment}, \mathsf{telescope}(\mathsf{optical}),\\ & \mathsf{available}(\mathsf{observatory}, \mathsf{telescope}(\mathsf{optical})\ \},\\ & \mathsf{performable}(\mathsf{dante}, \mathsf{experiment})\ \rangle\end{aligned}$$

There exist at least two different potential arguments for argument **a**, *each of which is distinct (i.e. neither argument is a potential argument or elaboration of the the other):*

$$\begin{aligned}\mathbf{b} = \langle\ \{\ & \mathsf{assignable}(\mathsf{dante},\mathsf{observatory}),\\ & \mathsf{requires}(\mathsf{experiment},\mathsf{telescope}(\mathsf{optical}),\\ & \mathsf{available}(\mathsf{observatory},\mathsf{telescope}(\mathsf{optical}))\ \},\\ & \mathsf{performable}(\mathsf{dante},\mathsf{experiment})\ \rangle\end{aligned}$$

$$\begin{aligned}\mathbf{c} = \langle\ \{\ & \mathsf{registered}(\mathsf{dante},\mathsf{observatory}),\\ & \mathsf{suitable}(\mathsf{observatory},\mathsf{experiment}),\\ & \mathsf{performable}(\mathsf{dante},\mathsf{experiment})\ \rangle\end{aligned}$$

Argument **c** *could just as easily be potentially another argument however:*

$$\begin{aligned}\mathbf{d} = \langle\ \{\ & \mathsf{registered}(\mathsf{dante},\mathsf{observatory}),\\ & \mathsf{type}(\mathsf{experiment},\mathsf{observation}),\\ & \mathsf{preferred}(\mathsf{observatory},\mathsf{observation}),\\ & \forall X,Y,Z.\ \mathsf{type}(X,Y)\wedge\mathsf{preferred}(Z,Y)\rightarrow\mathsf{suitable}(X,Z)\ \},\\ & \mathsf{performable}(\mathsf{dante},\mathsf{experiment})\ \rangle\end{aligned}$$

Arguments **a** *and* **d** *are themselves distinct.*

We can now define a *potential restriction* — a simplification of a set of arguments so as to fit within a particular argument space:

Definition 4.2 *A* **potential restriction** *of a set of arguments S into an argument space* Δ *within an argumentation framework* $(\mathcal{L},\vdash,\Delta)$ *is any set of potential arguments* S' *where:*

- *For each argument* $\mathbf{a}\in S'$*, it is the case that* $\mathbf{a}\in\Delta$ *and* $\mathbf{a}\sqsubseteq\mathbf{b}$ *for some argument* $\mathbf{b}\in S$*.*
- *For every argument* $\mathbf{b}\in S$*, if there exists no argument* $\mathbf{a}\in S'$ *such that* $\mathbf{a}\sqsubseteq\mathbf{b}$*, then there exists no argument* $\mathbf{c}\in\Delta$ *such that* $\mathbf{c}\sqsubseteq\mathbf{b}$*.*

By restricting arguments into a more constrained argument space, we can describe an agent's beliefs in such a way as to *potentially* match a number of more nuanced viewpoints. Even if those viewpoints are jointly inconsistent, they can be said to be in agreement within the hypothesis space described by the potential restriction.

Example 4.3 *Consider the following argument set:*

$$\begin{aligned}\mathbf{a} = \langle\ \{\ & \mathsf{tenure}(\mathsf{alanna},\mathsf{edinburgh}),\\ & \mathsf{university}(\mathsf{edinburgh}),\\ & \forall X,Y.\ \mathsf{tenure}(X,Y)\wedge\mathsf{university}(Y)\rightarrow\mathsf{researcher}(X),\\ & \neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),\end{aligned}$$

$\forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X,Y)\ \},$
$\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{b} = \langle\ \{\ \mathsf{postdoc}(\mathsf{alanna}),$
$\forall X.\ \mathsf{postdoc}(X) \rightarrow \mathsf{researcher}(X),$
$\mathsf{needs}(\mathsf{alanna},\mathsf{data}),$
$\mathsf{only\_source}(\mathsf{library},\mathsf{data}),$
$\forall X,Y,Z.\ \mathsf{needs}(X,Y) \wedge \mathsf{only\_source}(Z,Y) \rightarrow \neg\mathsf{beneficial}(X,\mathsf{abuse}(\mathsf{access}(Z))),$
$\forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X,Y)\ \},$
$\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{c} = \langle\ \{\ \mathsf{abused}(\mathsf{alanna},\mathsf{access}(\mathsf{laboratory})),$
$\mathsf{analogous}(\mathsf{access}(\mathsf{laboratory}),\mathsf{access}(\mathsf{library})),$
$\forall X,Y,Z.\ \mathsf{abused}(X,Y) \wedge \mathsf{analogous}(Y,Z) \rightarrow \neg\mathsf{trustworthy}(X,Z)\ \},$
$\neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{d} = \langle\ \{\ \neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),$
$\mathsf{analogous}(\mathsf{access}(\mathsf{laboratory}),\mathsf{access}(\mathsf{library})),$
$\forall X,Y,Z.\ \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \wedge \mathsf{analogous}(Z,Y)$
$\rightarrow \neg\mathsf{beneficial}(X,\mathsf{abuse}(Z))\ \},$
$\neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{laboratory})))\ \rangle$

$\mathbf{e} = \langle\ \{\ \mathsf{tenure}(\mathsf{alanna},\mathsf{edinburgh}),$
$\mathsf{university}(\mathsf{edinburgh}),$
$\forall X,Y.\ \mathsf{tenure}(X,Y) \wedge \mathsf{university}(Y) \rightarrow \mathsf{researcher}(X),$
$\neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),$
$\neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},$
$\mathsf{researcher}(\mathsf{alanna}) \wedge \neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))) \wedge$
$\neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{f} = \langle\ \{\ \neg\exists X.\ \mathsf{research\_topic}(\mathsf{alanna},X),$
$\forall X.\ \mathsf{researcher}(X) \rightarrow \exists Y.\ \mathsf{research\_topic}(X,Y)\ \},$
$\neg\mathsf{researcher}(\mathsf{alanna})\ \rangle$

$\mathbf{g} = \langle\ \{\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

Now consider an argument space Δ *wherein the horizon contains any sentence drawn from the set* $\{\mathsf{researcher}(X_1,X_2), \mathsf{beneficial}(X_3,X_4), \mathsf{abused}(X_5,X_6), \mathsf{analogous}(X_7,X_8)\}$ *or their negations, along with any implication which draws on that set to formulate its own antecedent; the focus of the argument space is on any instance or refutation of* $\mathsf{trustworthy}(X_9,X_{10})$. *The following revised set of arguments represents a potential restriction of the above argument set into* Δ*:*

$\mathbf{a}' = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}),$
$\neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),$
$\forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X,Y)\ \},$
$\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{c}' = \langle\ \{\ \mathsf{abused}(\mathsf{alanna},\mathsf{access}(\mathsf{laboratory})),$
$\mathsf{analogous}(\mathsf{access}(\mathsf{laboratory}),\mathsf{access}(\mathsf{library})),$
$\forall X,Y,Z.\ \mathsf{abused}(X,Y) \wedge \mathsf{analogous}(Y,Z) \rightarrow \neg\mathsf{trustworthy}(X,Z)\ \},$
$\neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{e}' = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}),$
$\neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),$
$\neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},$

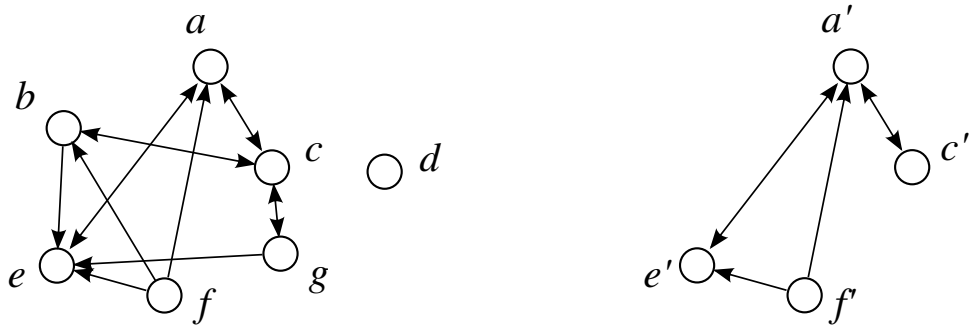


Figure 4.1: *On the left, the system of arguments described in Example 4.3 prior to restriction. On the right, the system of arguments after restriction.*

$$\begin{array}{l} \quad \mathsf{researcher}(\mathsf{alanna}) \wedge \neg \mathsf{beneficial}(\mathsf{alanna}, \mathsf{abuse}(\mathsf{access}(\mathsf{library}))) \wedge \\ \quad \neg \mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library})) \rangle \\ \mathbf{f}' = \langle\ \{\ \neg \exists X.\ \mathsf{researcher}(X)\ \},\ \neg \mathsf{researcher}(\mathsf{alanna}) \rangle \end{array}$$

Arguments **a** *and* **b** *both share the same potential argument* $\mathbf{a}'$. *Argument* **c** *fits into the argument space without modification. Argument* **d** *is not considered to be relevant (it does not claim an assertion in the focus, nor does it attack a relevant argument). Argument* **e** *is relevant because it attacks argument* $\mathbf{a}'$*, its claim being contrary to the sentence* $\forall X, Y.\ \mathsf{researcher}(X) \wedge \neg \mathsf{beneficial}(X, \mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X, Y)$ *in* $\mathbf{a}'$. *Argument* **f** *attacks arguments* $\mathbf{a}'$ *and* $\mathbf{e}'$. *Argument* **g** *is already too abstract for the argument space; in this case argument* $\mathbf{a}'$ *provides a suitably detailed replacement argument.*

Since for our portrayals of distributed interaction we ideally want to do just enough argumentation to identify any disputes as to the resolution of a given interaction constraint and then resolve them, portrayals should restrict the arguments given by agents into a quite limited argument space. In essence, we want to restrict any arguments so that they describe just enough about an agent's beliefs to support its claims whilst concealing anything which might be controversial, but which does not need to be expressed in order to differentiate between different resolutions of given constraints on interaction. We do *not* want to inadvertently suppress any valid resolutions of a constraint.

Alternatively, if we want to elaborate upon arguments as a group in order to fit them into a broader argument space, we can perform a *potential expansion*:

Definition 4.3 *A* **potential expansion** *of a set of arguments S to the extent of an argument space* Δ *within an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *is any set of elaborations*

S' where:

- *For each argument* $\mathbf{a} \in S'$*, it is the case that* $\mathbf{a} \in \Delta$ *and* $\mathbf{b} \sqsubseteq \mathbf{a}$ *for some argument* $\mathbf{b} \in S$*.*
- *For every argument* $\mathbf{b} \in S$*, if there exists no argument* $\mathbf{a} \in S'$ *such that* $\mathbf{b} \sqsubseteq \mathbf{a}$*, then there exists no argument* $\mathbf{c} \in \Delta$ *such that* $\mathbf{b} \sqsubseteq \mathbf{c}$*.*

If an argument can be elaborated upon to the extent that its support becomes wholly drawn from the argument space horizon, then it then can be said to be properly placed within that argument space.

Example 4.4 *Consider the following argument set:*

$$
\begin{aligned}
\mathbf{a} = \langle\ \{\ & \mathsf{certified}(\mathsf{alanna}),\\
& \neg\mathsf{at\_capacity}(\mathsf{library}),\\
& \forall X, Y.\ \mathsf{certified}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))\ \},\\
& \mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\ \rangle\\
\mathbf{b} = \langle\ \{\ & \mathsf{capacity}(\mathsf{library}, 200),\\
& \mathsf{user\_count}(\mathsf{library}, 193),\\
& \forall X, Y, Z.\ \mathsf{capacity}(X, Y) \wedge \mathsf{user\_count}(X, Z) \wedge Z < Y \rightarrow \neg\mathsf{at\_capacity}(X)\ \},\\
& \neg\mathsf{at\_capacity}(\mathsf{library})\ \rangle\\
\mathbf{c} = \langle\ \{\ & \neg\mathsf{researcher}(\mathsf{alanna})\ \},\ \neg\mathsf{researcher}(\mathsf{alanna})\ \rangle
\end{aligned}
$$

Now consider an argument space Δ *wherein the horizon can be described by the following formulae:*

$$
\begin{aligned}
& \mathsf{capacity}(X_1, X_2),\\
& \mathsf{user\_count}(X_3, X_4),\\
& \forall X, Y, Z.\ \mathsf{capacity}(X, Y) \wedge \mathsf{user\_count}(X, Z) \wedge Z < Y \rightarrow \neg\mathsf{at\_capacity}(X),\\
& \mathsf{employed}(X_5, X_6),\\
& \mathsf{backing}(X_7, X_8),\\
& \forall X.\ \mathsf{employed}(X, Y) \wedge \mathsf{backing}(X, Y) \rightarrow \mathsf{certified}(X),\\
& \neg\mathsf{published}(X_9),\\
& \forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{published}(X),\\
& \forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{certified}(X),\\
& \mathsf{at\_capacity}(X_{10}),\\
& \forall X, Y.\ \mathsf{certified}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))
\end{aligned}
$$

The focus of Δ *is on any instance or refutation of* $\mathsf{eligible}(X_{11}, X_{12})$*. The following revised set of arguments represents a potential expansion of our original argument set into that argument space:*

$$
\begin{aligned}
\mathbf{a}_1 = \langle\ \{\ & \mathsf{employed}(\mathsf{alanna}, \mathsf{edinburgh}),\\
& \mathsf{backing}(\mathsf{alanna}, \mathsf{edinburgh}),\\
& \forall X.\ \mathsf{employed}(X, Y) \wedge \mathsf{backing}(X, Y) \rightarrow \mathsf{certified}(X),\\
& \mathsf{at\_capacity}(\mathsf{library}),
\end{aligned}
$$

$\forall X, Y.\ \mathsf{certified}(X) \wedge \neg \mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))\ \},$
$\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$
$\mathbf{b}_1 = \langle\ \{\ \mathsf{capacity}(\mathsf{library}, 200),$
$\mathsf{user\_count}(\mathsf{library}, 193),$
$\forall X, Y, Z.\ \mathsf{capacity}(X, Y) \wedge \mathsf{user\_count}(X, Z) \wedge Z < Y \rightarrow \neg \mathsf{at\_capacity}(X)\ \},$
$\neg \mathsf{at\_capacity}(\mathsf{library})$

Argument $\mathbf{a}'$ *elaborates upon argument* **a***, whilst argument* **b** *does not require further expansion. Argument* **c** *is not relevant here, because its claim does not match the focus of* Δ*, nor does it attack* $\mathbf{a}'$ *or* **b***. However there is another potential expansion of the arguments above:*

$\mathbf{a}_2 = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}),$
$\forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{certified}(X),$
$\neg \mathsf{at\_capacity}(\mathsf{library}),$
$\forall X, Y.\ \mathsf{certified}(X) \wedge \neg \mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))\ \},$
$\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\ \rangle$
$\mathbf{b}_2 = \langle\ \{\ \mathsf{capacity}(\mathsf{library}, 200),$
$\mathsf{user\_count}(\mathsf{library}, 193),$
$\forall X, Y, Z.\ \mathsf{capacity}(X, Y) \wedge \mathsf{user\_count}(X, Z) \wedge Z < Y \rightarrow \neg \mathsf{at\_capacity}(X)\ \},$
$\neg \mathsf{at\_capacity}(\mathsf{library})\ \rangle$
$\mathbf{c}_2 = \langle\ \{\ \neg \mathsf{published}(\mathsf{alanna}),$
$\forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{published}(X)\ \},$
$\neg \mathsf{researcher}(\mathsf{alanna})\ \rangle$

In this case, the chosen elaboration of argument **a** *is attackable by argument* $\mathbf{c}_2$.

The above example illustrates an important point; depending upon how an expansion of a set of arguments is performed, there may be need to generate new arguments to adequately explore the greater argument space into which that argument was moved.

Of course an agent must be *able* to elaborate upon an argument — this requires knowledge about the provenance of propositions that goes beyond that already described in the original hypothesis space in which the original argument was created. It may well be that an agent has more knowledge about a given topic, but had felt it too detailed for the context in which a given set of beliefs is generated (i.e. the agent was abstracting its real knowledge in order to produce a simpler theory for solving the problem at hand, but later found it inadequate). Another option arises in social discourse — arguments are being expanded because new hypotheses have been introduced to an agent by a peer. In other words, the additional knowledge required is obtained from another peer, of presumably greater expertise in the domain in question. It is exactly this kind of scenario which we are interested in, and we shall see potential expansions of individual agents' argument spaces later in Chapter 5 as a side effect of dialogue with peers.

4.1.2 Rescoping Argumentation

Changes in the theory core can affect the interpretations of arguments, and thus the theory derived from an argument interpretation, as already discussed. In general, when a given assertion goes from known to unknown, we treat the assertion as if it was a generated hypothesis. Likewise, when a formerly hypothetical assertion becomes observed fact, it is added to the theory core. In either case, the argument space remains unchanged. It may be however that new information arises which was not previously given consideration even hypothetically, but which still might be considered pertinent to debate. Similarly, new hypotheses might be generated to explain certain phenomena in the environment. In either circumstance, it may be advantageous to extend the argument space, such that arguments can be constructed based on these new propositions:

Definition 4.4 *Given a set of hypotheses N, the argument space $\Delta = (H,F)$ of an assumption-based argumentation framework $(\mathcal{L},\vdash,\Delta)$ can be* **extended** *into a new space $\Delta' = (H',F)$ where:*

- *R is a set of sentences $\varphi \in H$ such that there exists a consistent subset $S \subseteq ((H \cup N)/R)$ such that $S \vdash \varphi$.*
- *N' is a set of sentences $\varphi \in N$ such that there does not exist a consistent subset $S \subseteq ((H \cup N)/N')$ such that $S \vdash \varphi$.*
- *The revised horizon $H' = (H/R) \cup N'$.*

An extended argument space is one where the horizon is redefined to accommodate additional sentences, allowing the production of arguments based on those sentences. Of course a system of arguments generated within the original argument space may no longer fit properly within the new space, and as such an agent will need to perform a potential expansion of arguments as per Definition 4.3:

Example 4.5 *Take the argument set at the beginning of Example 4.4. The horizon of the argument space Δ in which that set would constitute a valid system of arguments must include all assumptions used in arguments* **a**, **b** *and* **c**, *with the focus on* $\mathsf{eligible}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))$. *We can extend Δ to include the following sentences:*

$\mathsf{employed}(X_1,X_2)$,
$\mathsf{backing}(X_3,X_4)$,
$\forall X.\ \mathsf{employed}(X,Y) \wedge \mathsf{backing}(X,Y) \rightarrow \mathsf{certified}(X)$,
$\mathsf{published}(X_5)$,
$\neg\mathsf{published}(X_6)$,

$\forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{published}(X)$,
$\forall X.\ \mathsf{researcher}(X) \rightarrow \mathsf{certified}(X)$.

This revised argument space would exclude $\mathsf{certified}(X_7)$ *from the horizon (as it can be inferred from the new assertions added to the space) and insert into the horizon all of the new sentences other than* $\mathsf{published}(X_5)$ *(which is already derivable from the current space). This would then motivate either (or both) of the potential expansions described in Example 4.4.*

We have not addressed yet what happens to the attack relations between arguments in an argument system when those arguments are elaborated upon. Fundamentally, this is quite simple — when a set of arguments is expanded, all elaborations retain the attacks generated by or against them, because elaborations retain the same claim (and so still attack arguments containing assertions contrary to that claim) and an agent can derive from an elaboration all the results it could from the potential argument (and so can still be rebutted or undercut by the same attacks):

Theorem 4.1 *If* $\mathbf{a} \rightharpoonup \mathbf{b}$*, then both* $\mathbf{c} \rightharpoonup \mathbf{d}$ *and* $\mathbf{a} \rightharpoonup \mathbf{d}$*, if* $\mathbf{a} \sqsubseteq \mathbf{c}$ *and* $\mathbf{b} \sqsubseteq \mathbf{d}$*.*

Proof 4.1 *Within an assumption-based argumentation framework* $(\mathcal{L}, \vdash, \Delta)$*, it is the case that* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *if and only if* $\Psi \vdash \gamma$ *and* $\{\alpha\} \vdash \neg\gamma$ *for some logical sentence* $\gamma \in \mathcal{L}$ *as per Definition 3.10. We observe that:*

- *An elaboration of an argument* $\langle \Psi, \beta \rangle$ *is an argument* $\langle \Psi', \beta \rangle$ *wherein for every sentence* $\varphi \in \Psi$*, it is the case that* $\Psi' \vdash \varphi$*.*
- $\Psi' \vdash \gamma$ *by virtue of* $\Psi' \vdash \varphi$ *for all sentences in* $\varphi \in \Psi$ *and* $\Psi \vdash \gamma$*, and so any elaboration of the target argument of an attack will remain attacked after elaboration.*

Likewise, we observe that:

- *An elaboration of an argument* $\langle \Phi, \alpha \rangle$ *is an argument* $\langle \Phi', \alpha \rangle$ *wherein for every sentence* $\varphi \in \Phi$*, it is the case that* $\Phi' \vdash \varphi$*.*
- *The claim* α *is unchanged between an argument* $\langle \Phi, \alpha \rangle$ *and its elaboration* $\langle \Phi', \alpha \rangle$*, such that* $\langle \Phi', \alpha \rangle$ *continues to attack the same arguments as* $\langle \Phi, \alpha \rangle$*. Likewise, any argument attacked by* $\langle \Phi', \alpha \rangle$ *is attacked by its potential argument* $\langle \Phi, \alpha \rangle$*.*

Therefore, if $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *in an assumption-based argumentation framework, then* $\langle \Phi', \alpha \rangle \rightharpoonup \langle \Psi', \beta \rangle$ *and* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi', \beta \rangle$ *if* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Phi', \alpha \rangle$ *and* $\langle \Psi, \beta \rangle \sqsubseteq \langle \Psi', \beta \rangle$*.*

Of course, there may exist new attacks based on contrary relationships between assertions added to the horizon of the extended argument space; these attacks need to be identified and added to the argument system as normal for argument generation. It is also worth noting that if two arguments elaborate into the same argument (reflecting the situation in Example 4.2), then they cannot have attacked one another prior to elaboration:

Theorem 4.2 *If* $\mathbf{a} \sqsubseteq \mathbf{c}$ *and* $\mathbf{b} \sqsubseteq \mathbf{c}$*, then* $\mathbf{a} \not\rightharpoonup \mathbf{b}$.

Proof 4.2 *Given an assumption-based argumentation framework* $(\mathcal{L}, \vdash, \Delta)$*, if* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$*, but there exists an argument* $\mathbf{c}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{c}$ *and* $\langle \Psi, \beta \rangle \sqsubseteq \mathbf{c}$*, then:*

- *There exists a contrary relation* (α, γ) *such that* $\Psi \vdash \gamma$ *and* $\{\alpha\} \vdash \neg\gamma$ *(by Definition 3.10).*
- *For every sentence* $\varphi \in \Phi$ *and every sentence* $\varphi \in \Psi$*, it is the case that* φ *can be inferred from* $\mathbf{c}$ *(shown in Proof 4.1).*
- *Therefore, given that* $\Phi \vdash \alpha$*, from* $\mathbf{c}$ *we can infer* γ *and* γ*, in which case the support for* $\mathbf{c}$ *is internally inconsistent (as per Definition 3.4).*

Therefore $\mathbf{c}$ *is not a valid argument by Definition 3.10, and therefore if* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$*, then there cannot exist an argument* $\mathbf{c}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{c}$ *and* $\langle \Psi, \beta \rangle \sqsubseteq \mathbf{c}$.

By this proof can we have confidence that the elaboration of arguments will not interfere unduly with existing conflicts between arguments.

Instead of expanding an argument space, an agent may want to *contract* the argument space in which argumentation occurs. It might seem odd to want to do this, however whilst a larger argument space can potentially describe with greater accuracy a given problem domain, it also requires the generation of more arguments in order to adequately describe the space and ensure a consistent labelling of the sentences within, and it may be the case that this additional argumentation does not in fact have much influence on the conclusions which are of interest. For a given problem, it might be better to abstract aside detail, and work with simpler propositions in order to avoid further computation. Fundamentally, nothing short of a complete description of everything will guarantee perfect results, but in many practical problem domains, it is accepted that adequate (even very good) results can be obtained with less computation by using a more abstract depiction of the problem:

Definition 4.5 *The argument space $\Delta = (H, F)$ of an assumption-based argumentation framework $(\mathcal{L}, \vdash, \Delta)$ can be* **contracted** *into a new space $\Delta' = (H', F')$ excluding a set of sentences R where:*

- *$R' \subseteq H$ is a minimal set of sentences such that there exist no consistent subsets $S \subseteq (H/R')$ for which $S \vdash \varphi$, where $\varphi \in R$.*
- *The revised horizon $H' = (H/R')$.*
- *The revised focus $F' = (F/R)$.*

The deliberate use of smaller argument spaces really comes into its own in multi-agent argumentation, where several agents collaborate to produce a system of arguments. Since multi-agent argumentation is primarily engaged in in order to determine the best way to resolve some problem, there is little point in executing argumentation in an argument space which includes extraneous information. This means that the argument space in which multi-agent argumentation is executed will almost certainly be to varying degrees a contraction of the individual argument spaces in which agents decide their personal beliefs.

A contracted argument space is one in which certain assertions can no longer be inferred from the horizon including perhaps certain previously relevant conclusions. To fit into the new space, a potential restriction of the arguments used in an argument system may be necessary.

Example 4.6 *The potential restriction demonstrated by Example 4.3 could have been motivated by a contraction of the original argument space Δ in which the system of arguments described in that example was generated. This could be represented by the exclusion of instances of the set* $\{\ \mathsf{tenure}(X_1, X_2),\ \mathsf{university}(X_3),\ \mathsf{postdoc}(X_4),\ \mathsf{needs}(X_5, X_6),\ \mathsf{only\_source}(X_7, X_8),\ \mathsf{research\_topic}(X_9, X_{10})\ \}$ *from the space (a rather significant pruning), leading to the reduced system of arguments at the end of Example 4.3. It would also require a change in the focus of the revised argument space, such that argument* **d** *(which claims* $\neg\mathsf{beneficial}(\mathsf{alanna}, \mathsf{abuse}(\mathsf{access}(\mathsf{laboratory}))))$ *becomes irrelevant.*

When a set of arguments is restricted, all potential arguments retain the same claim (and so still attack arguments containing contrary assertions), but any attacks against results which can no longer be derived from the potential arguments (i.e. attacks against supporting assumptions which have been discarded in favour of derivative assumptions) are lost from the argument system. Naturally, this can change the acceptability

of arguments in an argument system, illustrating if nothing else the danger in abstracting aside information. Nonetheless, there are notions we can employ to describe 'good' argument spaces, which we can make use of.

The risk inherent in moving arguments into a more limited argument space is that the common dependencies of arguments may be concealed, hiding the fact that an attack against one argument may also be an attack against others. Consider two arguments, both of which share a common assumption. Now consider the insertion of these arguments into an argument space, where perhaps they must be potentially restricted in order to fit. It is important that either the common assumption is not in either restricted argument and is not inferable from either support set (in which case the common assumption is simply out of scope, and irrelevant), or that the common assumption is inferable from *both* argument support sets. Should the assumption be inferable from one, but not the other, then should that assumption or its derivatives be contradicted by a new argument, only one of the two original arguments will be formally attacked, which may lead to an inaccurate evaluation of the argument system.

If a common premise falls within the argument space of an argumentation framework, then it is important that it is inferable (trivially or otherwise) from the support sets of *all* arguments which depend on it. The easiest way to ensure this is to exploit the horizon of the argument space, since all arguments already within it are founded on it:

Definition 4.6 *A system of arguments* $(\mathcal{A}, \rightharpoonup)$ *is* **balanced** *within an argument space* Δ *with respect to a theory context* $\mathcal{C}$ *with an accepted extension* $\mathcal{E}$ *if and only if:*

- $\mathcal{A} \subseteq \Delta$ *(i.e. for every argument* $\mathbf{a} \in \mathcal{A}$*, it is the case that* $\mathbf{a} \in \Delta$*).*
- *There exist no two arguments* $\langle \Phi, \alpha \rangle, \langle \Psi, \beta \rangle \in \mathcal{A}$ *such that for* all *elaborations* $\langle \Phi', \alpha \rangle \in \mathcal{E}$ *of* $\langle \Phi, \alpha \rangle$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Phi', \alpha \rangle$*, there exists a common dependency* φ *such that* $\Psi \vdash \varphi$ *and* $\Phi' \vdash \varphi$*, but* $\Phi \vdash \varphi$*.*

If $(\mathcal{A}, \rightharpoonup)$ *is unbalanced within* Δ *with respect to* $\mathcal{C}$*, but* $\mathcal{A} \subseteq \Delta$*, then* Δ *itself is* **ill-formed** *with respect to* Π*, where* Π *is the theory derived from* $\mathcal{C}$*.*

In essence, all related arguments should be expressed to the same level of detail. Note that it is meaningless to evaluate the balance of a system of arguments without an external reference — therefore argument balance is of most concern for agents trying to express their theories in a more restricted space than the ones in which they formed

those theories (e.g. a shared argument space created by dialogue between heterogeneous agents). Note also that in Definition 4.6, we only insist that a system of arguments be balanced from the perspective of *accepted* arguments generated in a greater argumentation framework, rather than any and all arguments. This is because different interpretations of that greater framework may promote different dependencies for arguments in the restricted space, and applying even those dependencies an agent does not itself believe in would be overly demanding (and possibly even unjustifiable for inadmissible arguments). Finally, note that if an agent accepts two alternative elaborations of one potential argument, and a dependency is present only in one of those elaborations, then the potential argument need not be expanded to include that dependency, because it can 'potentially' be the other elaboration.

Example 4.7 *Consider the following system of arguments:*

$$\begin{aligned}
\mathbf{a} = \langle\ \{\ & \mathsf{researcher}(\mathsf{alanna}),\\
& \neg\mathsf{at_capacity}(\mathsf{library}),\\
& \forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{at_capacity}(Y) \rightarrow \mathsf{eligible}(X,\mathsf{access}(Y))\ \},\\
& \mathsf{eligible}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle\\
\mathbf{b} = \langle\ \{\ & \mathsf{speciality}(\mathsf{alanna},\mathsf{argumentation}),\\
& \mathsf{lecturer}(\mathsf{alanna}),\\
& \forall X,Y.\ \mathsf{speciality}(X,Y) \wedge \mathsf{lecturer}(X) \rightarrow \mathsf{expert}(X,Y)\ \},\\
& \mathsf{expert}(\mathsf{alanna},\mathsf{argumentation})\ \rangle\\
\mathbf{c} = \langle\ \{\ & \mathsf{undergraduate}(\mathsf{alanna}),\\
& \forall X.\ \mathsf{undergraduate}(X) \wedge \mathsf{researcher}(X) \rightarrow \mathsf{false}\ \},\\
& \neg\mathsf{researcher}(\mathsf{alanna})\ \rangle
\end{aligned}$$

Assume that there exists an agent which accepts one and only one elaboration of argument **b** *within its theory context:*

$$\begin{aligned}
\mathbf{b}_1 = \langle\ \{\ & \mathsf{speciality}(\mathsf{alanna},\mathsf{argumentation}),\\
& \mathsf{researcher}(\mathsf{alanna}),\\
& \mathsf{teacher}(\mathsf{alanna}),\\
& \forall X,Y.\ \mathsf{researcher}(X) \wedge \mathsf{teacher}(X) \rightarrow \mathsf{lecturer}(X),\\
& \forall X,Y.\ \mathsf{speciality}(X,Y) \wedge \mathsf{lecturer}(X) \rightarrow \mathsf{expert}(X,Y)\ \},\\
& \mathsf{expert}(\mathsf{alanna},\mathsf{argumentation})\ \rangle
\end{aligned}$$

Because $\mathsf{researcher}(\mathsf{alanna})$ *can be derived from the support of argument* $\mathbf{b}_1$*, which elaborates upon argument* **b***, but* $\mathsf{researcher}(\mathsf{alanna})$ *is not found in* **b** *despite that assertion having to be part of the horizon of any argument space* Δ *which can generate arguments* **a**, **b** *and* **c***, the system of arguments consisting of arguments* **a**, **b** *and* **c** *is unbalanced within its argument space from the perspective of the given agent's theory context.*

In this case, the unbalanced nature of the argument system conceals the possibility that argument **c** *attacks argument* **b** *as well as argument* **a** *(whether this is borne out or not depends on the ability of any contributing agent to produce an alternative elaboration of argument* **c** *which does not rely on* $\{a\}$*). Put in another way, there exists from the perspective of our agent an inclination to consider the base assumption* $\mathsf{lecturer}(\mathsf{alanna})$ *to be partially dependent on* $\mathsf{researcher}(\mathsf{alanna})$*, thus suggesting to it that the current horizon of* Δ *could perhaps be modified to be less inter-dependent (and thus make the base assumptions from any derived theory more independent from one another).*

If an argument space is ill-formed from the perspective of a given outside theory context, then the horizon of the space contains assertions which can be at least partially inferred from other assertions within it (because an argument's support is built from assertions in the horizon, and part of the support for one argument can be partially inferred from the support of another). It can be proven that there exists an expansion of the space which, along with a potential expansion of the arguments within it, would result in a balanced system of arguments from the perspective of that outside theory context to which the argument system is compared:

Theorem 4.3 *If an argument space* Δ *is ill-formed with respect to context* $\mathcal{C}$*, then there exists an expansion* Δ' *of* Δ *which* is *well-formed with respect to* $\mathcal{C}$*.*

Proof 4.3 *If an argument space* $\Delta = (H, F)$ *is ill-formed with respect to a theory context* $\mathcal{C}$*, then:*

- *There exists an argument* $\langle \Phi, \alpha \rangle \in \Delta$ *which is potentially an argument* $\langle \Psi, \alpha \rangle \in \mathcal{E}$*, where* $\mathcal{E}$ *is the accepted extension of* $\mathcal{C}$ *such that* $\Psi \vdash \varphi$ *for some* $\varphi \in H$*.*
- *There exists no other argument* $\langle \Psi', \alpha \rangle \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi', \alpha \rangle$ *and* $\Psi' \nvdash \varphi$*.*
- $\varphi \notin \Phi$*.*

This means that for some sentence $\varphi \in H$*, there exists an internally consistent set of sentences* $S \subseteq ((H \cup \Pi)/\{\varphi\})$ *such that* $S \vdash \varphi$ *and* $(S \cap H) \neq \emptyset$ *(i.e.* φ *is at least partially derivable from H), where* Π *is the theory derivable from the accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$*. We observe that there exists a 'maximal' argument space* $\Delta^+ = (H^+, F)$ *such that* $H^+ = (H \cup \Pi)$*:*

- *We can iteratively reduce Δ^+ by finding sentences $\varphi \in H^+$ for which there exists an internally consistent subset $S \subseteq (H^+/\{\varphi\})$ such that $S \vdash \varphi$ and then removing φ from H^+.*
- *Every sentence $\varphi \in H^+$ removed can be inferred from S; any sentence $\psi \in S$ likewise removed later must also be derivable from an internally consistent subset of the remainder of H^+, so no deductive capacity is lost.*

If there is no sentence φ left in H^+ for which there exists an internally consistent set of sentences $S \subseteq (H^+/\{\varphi\})$ such that $S \vdash \varphi$, then given that for every sentence $\psi \in \Pi$, there exists an internally consistent subset $S_2 \subseteq H^+$ such that $S_2 \vdash \psi$:

- *There must be no sentence $\varphi \in H^+$ for which there exists an internally consistent set of sentences $S \subseteq ((H^+ \cup \Pi)/\{\varphi\})$ such that $S \vdash \varphi$ and $(S \cap H^+) \neq \emptyset$.*
- *There exists no argument $\langle \Phi, \alpha \rangle \in \Delta^+$ which is potentially an argument $\langle \Psi, \alpha \rangle \in \mathcal{E}$ such that $\Psi \vdash \varphi$ for any $\varphi \in H^+$.*

Therefore there exists an expansion of Δ which is well-formed with respect to Π.

By this proof can we be confident that it will always be possible to achieve a balanced argument system by elaborating upon arguments — such a process will always terminate. This result can be exploited to refine the definition of an argument space — for example:

Example 4.8 *Consider an argumentation framework $(\mathcal{L}, \vdash, \Delta)$ wherein the horizon of Δ includes the sentences:*

$$\begin{array}{l} \mathsf{researcher}(\mathsf{alanna}), \\ \neg\mathsf{at\_capacity}(\mathsf{library}), \\ \mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}) \end{array}$$

Now consider the following argument:

$$\begin{array}{l} \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}), \\ \quad \neg\mathsf{at\_capacity}(\mathsf{library}), \\ \quad \forall X, Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathit{access}(Y))\ \}, \\ \ \mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library})\ \rangle \end{array}$$

This argument states that $\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ *can be derived from the horizon of Δ if it is extended to include the rule* $\forall X, Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))$ *(which would also lead to* $\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ *being rmoved), suggesting that with such an extension, the argument space would be better able to describe the domain of argumentation (on the basis that the new horizon then describes a better, flatter axiomatisation of that hypothesis space).*

Whilst in general argument balancing is done with respect to a given outside theory context (e.g. for evaluating a social argument space from the perspective of an agent's own beliefs), it is possible to evaluate the balance of a system of arguments within a theory context. In this case, Definition 4.6 serves to identify where elements of the horizon of an argument space are derivable from other elements of the horizon (in which case the argumentation framework is *not* a flat framework as defined in [Dung et al., 2007]).

The ability to compare argument spaces, and the systems of arguments which might be generated from those spaces is very important to this thesis, because this ability allows us to relate any shared system of arguments, created as part of a multi-agent argumentation process, with the corresponding internal arguments used by an agent to construct its standing beliefs. Moreover, the shared argument system becomes an avenue by which we can compare the internal arguments of *all* the autonomous agents involved in argumentation. In order to demonstrate this further though, we need to further consider the practicalities of multi-agent argumentation, as well as its theoretic qualities.

4.2 Agent Belief Synchronisation

Perhaps the primary purpose of social argumentation, from a hypothesis generation perspective, is that it provides justification to extend individual argument spaces to include new assumptions and thus produce new arguments and new interpretations of existing arguments. It is not that we expect social argumentation to be conducted from the outset within the intersection of agents' individual argument spaces, it is that we expect that *after* social argumentation has been conducted, that the arguments produced will now be found in some form or another within every participant's expanded theory context.

Basically, every time an entirely new argument is encountered by an agent, we expect that the agent revises the programme by which it determines its own beliefs such that it accounts for the hypotheses used by that argument. If every agent is able to factor every argument produced in a social argumentation process into their theory contexts, then each defeasible theory derived should be admissible with respect to the combined wisdom of the agent group, and thus will either be broadly compatible with every other theory, or will at least represent a rational counter-interpretation.

It is vital that we stop to consider what it is agents want to achieve when they

engage in social argumentation. Superficially, there are a number of reasons why an agent might engage in dialogue with its peers; it might want information, it might want to enlist the assistance of peers for some task, it might want to negotiate some kind of common contract for future services or behaviour. Underlying all this however is the observation that all agents still possess their own internal beliefs, and ultimately all agents will act according to those beliefs. Argumentation is only meaningful where it influences those beliefs; any external system of arguments will only be useful insofar as it might be expected in some way to reflect the internal argument systems of its contributors. The core purpose of social argumentation then is to bring the beliefs of agents into greater alignment.

Ideally, a group of agents would enter argumentation focused on a particular problem and at the end of the argumentation process, all agents would come to a common consensus as to the correct interpretation of the arguments articulated. In practice however, we already know that a system of arguments, even when it exhaustively explores a given argument space, may still have more than one admissible interpretation. Selection of a particular interpretation is then to some extent arbitrary given that any purely logical selection mechanism would be able to be articulated as a set of conclusive arguments itself. Thus, despite the best efforts of agents, a shared system of arguments may remain ambiguous. We must also consider agent autonomy. No outside entity has an inherent right to dictate a particular interpretation of arguments to an autonomous agent when there exist other just as valid interpretations available (unless the agent has voluntarily conceded responsibility for that decision). Whilst there may be circumstances where interaction is engaged in specifically to produce a common interpretation of a contentious concern, that is merely a matter of social commitment — what an agent fundamentally *believes* can only be determined by the agent itself. Thus there will be circumstances where no agreement between agents can be reached through no failing of the process by which argumentation was conducted. If this is the case, then how do we evaluate a social argumentation process?

For our purposes, to *synchronise* two theories is to ensure that they both represent two acceptable interpretations of the same body of evidence:

Definition 4.7 *A set of theories* $\Pi_1, \ldots, \Pi_n$ *(where* $n > 1$*) is* **synchronised** *given a theory core* Θ *if there exists a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *generated within an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *such that:*

- *For every theory* Π_i *there exists an admissible extension* $\mathcal{E}_i \subseteq \mathcal{A}$ *such that* $\Pi_i \subseteq$

$(\Theta \cup \bigcup_{\langle \Phi, \alpha \rangle \in \mathcal{E}_i} \Phi)$.

If two theories are synchronised, then those theories are both equally admissible given the evidence available. Any conflict between the theories is a product of interpretative bias in the absence of additional information, and as such can only be confidently resolved by further observation. It is *not* necessary for the two theories to be jointly consistent. Synchronised theories can be generated independently — the argumentation framework merely acts as a context in which the theories can be compared. Thus, synchronisation is something which should be viable in a multi-agent context.

There is however a weakness to this notion of synchronisation. It is in fact very difficult to synchronise *all* agent beliefs — not in fact much easier than ensuring consistency between all agent beliefs. Naturally, it would be preferable if we only looked at synchronisation of agent theories within the argument space of a particular social argumentation process:

Definition 4.8 *A set of theories $\Pi_1, \ldots, \Pi_n$ (where $n > 1$) is* **synchronised** *within an argument space Δ if and only if for each theory Π_i (where $1 \leq i \leq n$):*

- *There exists a system of arguments $(\mathcal{A}, \rightharpoonup)_i$ generated within an argumentation framework $(\mathcal{L}, \vdash, \Delta)_i$ such that Π_i is derived from a complete extension $\mathcal{E}_i$ of $(\mathcal{A}, \rightharpoonup)_i$ in accordance with a theory core Θ_i.*
- *There exists a potential restriction $\mathcal{E}'_i$ of $\mathcal{E}_i$ into Δ.*
- *For each theory Π_j (where $1 \leq j \leq n$), $\mathcal{E}'_i$ is a potential restriction into Δ of an admissible extension of argument system $(\mathcal{A}, \rightharpoonup)_j$.*

Thus the beliefs of agents are synchronised within a given argument space if the arguments generated within that space from one agent's beliefs cannot be shown to be inadmissible using arguments generated from the beliefs of its peers. The beliefs of agents may not be synchronised *outside* of that space, but any arguments which can be generated from those beliefs beyond the scope of the given argument space are considered irrelevant.

Example 4.9 *Consider the following system of arguments:*

$$\begin{aligned}\mathbf{a}_1 = \langle \{ & \mathsf{graduated}(\mathsf{alanna}),\\ & \mathsf{employed}(\mathsf{alanna}, \mathsf{edinburgh}),\\ & \mathsf{university}(\mathsf{edinburgh}),\\ & \forall X, Y.\ \mathsf{graduated}(X) \wedge \mathsf{employed}(X, Y) \wedge \mathsf{university}(Y) \rightarrow \mathsf{researcher}(X),\\ & \neg \mathsf{at\_capacity}(\mathsf{library}).\end{aligned}$$

$\forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X,\mathsf{access}(Y))\ \},$
$\mathsf{eligible}(\mathsf{alanna},\mathsf{library})\ \rangle$

$\mathbf{b}_1 = \langle\ \{\ \mathsf{undergraduate}(\mathsf{alanna}),$
$\forall X.\ \mathsf{undergraduate}(X) \wedge \mathsf{researcher}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{researcher}(\mathsf{alanna})\ \rangle$

$\mathbf{c}_1 = \langle\ \{\ \mathsf{graduated}(\mathsf{alanna}),$
$\forall X.\ \mathsf{graduated}(X) \wedge \mathsf{undergraduate}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{undergraduate}(\mathsf{alanna})\ \rangle$

$\mathbf{d}_1 = \langle\ \{\ \mathsf{undergraduate}(\mathsf{alanna}),$
$\forall X.\ \mathsf{graduated}(X) \wedge \mathsf{undergraduate}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{graduated}(\mathsf{alanna})\ \rangle$

$\mathbf{e}_1 = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}),$
$\neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),$
$\forall X,Y.\ \mathsf{researcher}(X) \rightarrow \mathsf{trustworthy}(X,\mathsf{access}(Y)) \vee$
$\mathsf{beneficial}(X,\mathsf{abuse}(\mathsf{access}(Y)))\ \},$
$\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{f}_1 = \langle\ \{\ \mathsf{source}(\mathsf{library},\mathsf{data}),$
$\mathsf{sellable}(\mathsf{alanna},\mathsf{eliza},\mathsf{data}),$
$\forall X,Y,Z,A.\ \mathsf{source}(X,Y) \wedge \mathsf{sellable}(Z,A,Y)$
$\rightarrow \mathsf{beneficial}(Z,\mathsf{abuse}(\mathsf{access}(X)))\ \},$
$\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library})))\ \rangle$

$\mathbf{g}_1 = \langle\ \{\ \mathsf{source}(\mathsf{laboratory},\mathsf{data}),$
$\mathsf{sellable}(\mathsf{benjamin},\mathsf{dante},\mathsf{data}),$
$\neg\mathsf{beneficial}(\mathsf{benjamin},\mathsf{abuse}(\mathsf{access}(\mathsf{laboratory})))\ \},$
$\mathsf{source}(\mathsf{laboratory},\mathsf{data}) \wedge \mathsf{sellable}(\mathsf{benjamin},\mathsf{dante},\mathsf{data}) \wedge$
$\neg\mathsf{beneficial}(\mathsf{benjamin},\mathsf{abuse}(\mathsf{access}(\mathsf{laboratory})))\ \rangle$

Assume that this describes agent Alanna's beliefs, such that a theory can be derived from the admissible extension $\{\mathbf{a}_1,\mathbf{c}_1,\mathbf{e}_1,\mathbf{g}_1\}$. *Now consider another system of arguments:*

$\mathbf{a}_2 = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}),$
$\neg\mathsf{at\_capacity}(\mathsf{library}).$
$\forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{at\_capacity}(Y) \rightarrow \mathsf{eligible}(X,\mathsf{access}(Y))\ \},$
$\mathsf{eligible}(\mathsf{alanna},\mathsf{library})\ \rangle$

$\mathbf{b}_2 = \langle\ \{\ \mathsf{student}(\mathsf{alanna},\mathsf{edinburgh},\mathsf{artificial\_intelligence}),$
$\mathsf{degree}(\mathsf{artificial\_intelligence},\mathsf{edinburgh},\mathsf{bachelor}(\mathsf{science})),$
$\forall X,Y,Z,A.\ \mathsf{student}(X,Y,Z) \wedge \mathsf{degree}(Z,Y,\mathit{bachelor}(A)) \rightarrow \mathsf{undergraduate}(X),$
$\forall X.\ \mathsf{undergraduate}(X) \wedge \mathsf{researcher}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{researcher}(\mathsf{alanna})\ \rangle$

$\mathbf{c}_2 = \langle\ \{\ \mathsf{graduated}(\mathsf{alanna}),$
$\forall X.\ \mathsf{graduated}(X) \wedge \mathsf{undergraduate}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{undergraduate}(\mathsf{alanna})\ \rangle$

$\mathbf{d}_2 = \langle\ \{\ \mathsf{undergraduate}(\mathsf{alanna}),$
$\forall X.\ \mathsf{graduated}(X) \wedge \mathsf{undergraduate}(X) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{graduated}(\mathsf{alanna})\ \rangle$

$\mathbf{e}_2 = \langle\ \{\ \mathsf{undergraduate}(\mathsf{alanna}),$
$\forall X.\ \mathsf{undergraduate}(X) \wedge \mathsf{eligible}(X,\mathsf{access}(\mathsf{library})) \rightarrow \mathsf{false}\ \},$
$\neg\mathsf{eligible}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$

$\mathbf{f}_2 = \langle \{ \text{eligible}(\text{alanna}, \text{access}(\text{library})),$
$\forall X.\ \text{undergraduate}(X) \wedge \text{eligible}(X, \text{access}(\text{library})) \rightarrow \text{false} \},$
$\neg\text{undergraduate}(\text{alanna}) \rangle$
$\mathbf{g}_2 = \langle \{ \text{eligible}(\text{alanna}, \text{access}(\text{library})),$
$\text{undergraduate}(\text{alanna}) \},$
$\text{eligible}(\text{alanna}, \text{access}(\text{library})) \wedge \text{eligible}(\text{alanna}, \text{access}(\text{library})) \rangle$

In this case, assume that this describes agent Benjamin's beliefs, such that a theory can be derived from the admissible extension $\{\mathbf{b}_2, \mathbf{d}_2, \mathbf{g}_2\}$. *There exists an argument space* Δ *such that both arguments sets share the same potential restriction:*

$\mathbf{a}' = \langle \{ \text{researcher}(\text{alanna}),$
$\neg\text{at\_capacity}(\text{library}).$
$\forall X, Y.\ \text{researcher}(X) \wedge \neg\text{at\_capacity}(Y) \rightarrow \text{eligible}(X, \text{access}(Y)) \},$
$\text{eligible}(\text{alanna}, \text{library}) \rangle$
$\mathbf{b}' = \langle \{ \text{undergraduate}(\text{alanna}),$
$\forall X.\ \text{undergraduate}(X) \wedge \text{researcher}(X) \rightarrow \text{false} \},$
$\neg\text{researcher}(\text{alanna}) \rangle$
$\mathbf{c}' = \langle \{ \text{graduated}(\text{alanna}),$
$\forall X.\ \text{graduated}(X) \wedge \text{undergraduate}(X) \rightarrow \text{false} \},$
$\neg\text{undergraduate}(\text{alanna}) \rangle$
$\mathbf{d}' = \langle \{ \text{undergraduate}(\text{alanna}),$
$\forall X.\ \text{graduated}(X) \wedge \text{undergraduate}(X) \rightarrow \text{false} \},$
$\neg\text{graduated}(\text{alanna}) \rangle$

More importantly, each agent's accepted extension is admissible within this restriction, and each extension is a potential restriction of an admissible extension in both of the original argument systems, in spite of their differences.

An argument space is *sufficiently expressive* with respect to a given theory if the theory is able to defend itself within the space as well as it could outside of it. In other words, if an argument within the space is supported by hypotheses which are part of the given theory is then attacked by another argument in the space *and* it is possible to formulate a counter-argument from the theory, then it should be possible to construct that counter-argument or a potential argument for it within the argument space:

Definition 4.9 *The argument space* Δ *of an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *is* **sufficiently expressive** *with respect to a theory* Π *if and only if:*

- *There exists a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *such that* Π *can be derived from an admissible extension* $\mathcal{E}$ *of* $(\mathcal{A}, \rightharpoonup)$.

- *There exists a potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}$ *into* Δ.

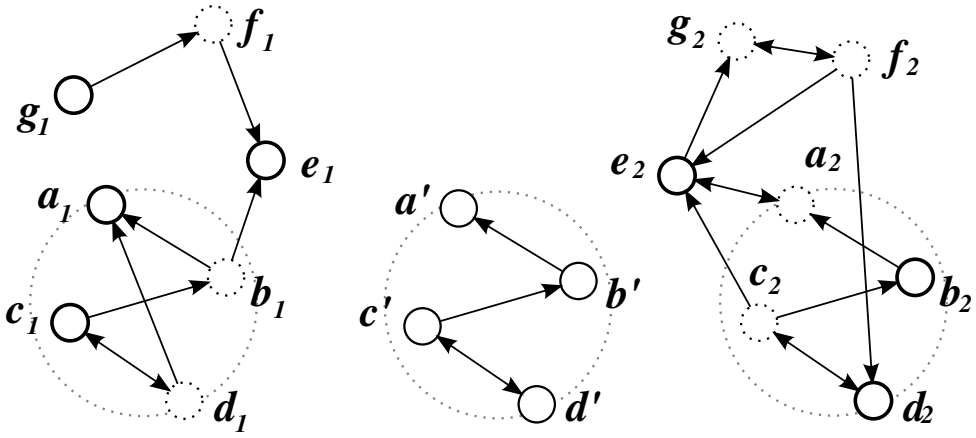


Figure 4.2: *The theories derived from the highlighted interpretations of the argument systems on the left and right are synchronised, because their potential restrictions into the argument system in the center are both admissible within the other argument system.*

- *If there exists an argument* $\mathbf{a} \in \Delta$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}'$*, then provided that there exists an argument* $\mathbf{c} \in \mathcal{E}$ *such that* $\mathbf{c} \rightharpoonup \mathbf{a}$*, there exists an argument* $\mathbf{d} \in \Delta$ *such that* $\mathbf{d} \sqsubseteq \mathbf{c}$ *and* $\mathbf{d} \rightharpoonup \mathbf{a}$*.*

If the argument space in which argumentation is conducted is found to be insufficiently expressive, then it can be expanded in order to ensure that the theories generated within it adequately approximate theories generated without. Of course, this only applies to circumstances where there exists external references by which to evaluate the arguments generated within a space; if a theory is being created wholly within an argument space, then there is no way to measure the quality of an argument space:

Theorem 4.4 *An argument space* Δ *of an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *is always sufficiently expressive with respect to a theory* Π *if* Π *is determined using a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *generated within* $(\mathcal{L}, \vdash, \Delta)$*.*

Proof 4.4 *From Definition 3.12, we see that a theory* Π *is derived from a theory core* Θ *and the resulting accepted extension* $\mathcal{E}$ *of a system of argument* $(\mathcal{A}, \rightharpoonup)$ *generated within the argument space* Δ *of an argumentation framework* $(\mathcal{L}, \vdash, \Delta)$ *such that* $\Pi = \Theta \cup (\bigcup_{\langle \Phi, \alpha \rangle \in \mathcal{E}} \Phi)$*. If* Δ *is* not *sufficiently expressive with respect to* Π*, then either:*

- *There exists no system of arguments* $(\mathcal{A}, \rightharpoonup)$ *such that* Π *can be derived from an admissible extension* $\mathcal{E}$ *of* $(\mathcal{A}, \rightharpoonup)$*.*

- *There exists no potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}$ *into* Δ.

- *There exists an argument* $\mathbf{a} \in \Delta$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}'$ *and every elaboration* $\mathbf{c} \in \Delta$ *for which* $\mathbf{a} \sqsubseteq \mathbf{c}$ *is defended against by an argument* $\mathbf{d} \in \mathcal{E}$ *such that* $\mathbf{d} \rightharpoonup \mathbf{c}$*, but there exists no argument* $\mathbf{e} \in \Delta$ *such that for at least one of those defenders* $\mathbf{d}$*, it is the case that* $\mathbf{e} \sqsubseteq \mathbf{d}$ *and* $\mathbf{e} \rightharpoonup \mathbf{a}$.

The first case is evidently false. As for the second case, it can quickly be seen that if $\mathcal{E} \subseteq \Delta$ *(i.e. all arguments in* $\mathcal{E}$ *are within* Δ*), then* $\mathcal{E}$ *is a potential restriction of* $\mathcal{E}$ *into* Δ*:*

- *For each argument* $\mathbf{a} \in \mathcal{E}$*, it is the case that* $\mathbf{a} \in \Delta$ *(because* $\mathcal{E} \subseteq \Delta$*) and* $\mathbf{a} \sqsubseteq \mathbf{a}$ *(by Definition 4.1).*

For the third case, we observe that any argument $\mathbf{d}$ *as described above must be within* Δ*, and therefore there is always a potential argument* $\mathbf{e} \in \Delta$ *such that* $\mathbf{e} \sqsubseteq \mathbf{d}$*, being* $\mathbf{d}$ *itself. It can be seen that* $\mathbf{d} \rightharpoonup \mathbf{a}$*, because* $\mathbf{d}$ *attacks an elaboration* $\mathbf{c}$ *of* $\mathbf{a}$*, which within* Δ *must be* $\mathbf{a}$ *itself. Therefore, an argument space is always sufficiently expressive for any theory derived from arguments generated within itself.*

If agents are using social argumentation to synchronise their beliefs, then they will likely only succeed within the social argument space if that space allows an agent to defend any of its beliefs which happen to fall within that space against attacks in that space which it would be able to defend against outside of it. If this is not the case, then it might not be possible to properly articulate an agent's beliefs during the argumentation process, in which case synchronisation will not occur unless all synchronising peers happen to be independently aware of the prohibited arguments anyway.

Theorem 4.5 *If an argument space* Δ *is not sufficiently expressive with respect to* Π*, then it cannot be certain that* Π *is synchronised with other theories* Π *within* Δ.

Proof 4.5 *If an argument space* Δ *is* not *sufficiently expressive with respect to* Π*, then either:*

- *There exists no system of arguments* $(\mathcal{A}, \rightharpoonup)$ *such that* Π *can be derived from an admissible extension* $\mathcal{E}$ *of* $(\mathcal{A}, \rightharpoonup)$.

- *There exists no potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}$ *into* Δ.

- *There exists an argument* $\mathbf{a} \in \Delta$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}'$ *and every elaboration* $\mathbf{c} \in \Delta$ *for which* $\mathbf{a} \sqsubseteq \mathbf{c}$ *is defended against by an argument* $\mathbf{d} \in \mathcal{E}$ *such that* $\mathbf{d} \rightharpoonup \mathbf{c}$*, but there exists no argument* $\mathbf{e} \in \Delta$ *such that for at least one of those defenders* $\mathbf{d}$*, it is the case that* $\mathbf{e} \sqsubseteq \mathbf{d}$ *and* $\mathbf{e} \rightharpoonup \mathbf{a}$*.*

The first case is evidently false. It also evident that there is always a potential restriction of a given set of arguments into an argument space, though that restriction may be the empty set if there is no overlap between the space in which the argument set was generated and the space into which it is restricted, so the second case must be false as well. As for the third case, if an argument $\mathbf{b} \in \mathcal{E}$ *is attacked by an argument* $\mathbf{a} \in \Delta$ *and every elaboration* $\mathbf{c}$ *of* $\mathbf{a}$ *is itself attacked by an argument* $\mathbf{d} \in \mathcal{E}$*, but there is no potential argument* $\mathbf{e} \in \Delta$ *of any* $\mathbf{d}$*, then* $\mathbf{b}$ *will be inadmissible within the argument system of* Δ*. Thus, unless all other agents consider an elaboration of* $\mathbf{b}$ *to be admissible in their own theory contexts, it is unlikely that a potential expansion of* $\mathcal{E}'$ *will be admissible in every one of those peers' theory contexts, and thus their theories will not be synchronised.*

Oddly enough, it would seem that any space with which a theory has no interaction is sufficiently expressive if attacks against that theory cannot be expressed there either. Of course, such a space is not *useful* from the perspective of that theory. Likewise, it is trivial to synchronise theories in a hypothesis space in which they do not interact with one another, but there is no benefit conferred from doing so (this is actually an important point for our contribution; we want to minimise the amount of argumentation we do in order to make computation easier and synchronisation easier to obtain, but we also want to maximise the benefit of doing so).

We have established the position that the purpose of social argumentation is to permit the synchronisation of agent beliefs within the argument space of the social interaction. At the very least, we can say that that is the purpose of social argumentation from the truth maintenance perspective, though there will also exist the purpose for which the argumentation process was instigated from the outset — such as creating a social contract for negotiation or planning. If the social system of arguments is merely an intermediary between the personal defeasible reasoning processes of individual agents however, then it should be possible to achieve synchronisation of agent theories via that argument system.

A system of arguments is considered to be *reconciled* with an agent's theory context if and only if the accepted beliefs of the agent are admissible within that argument

system and every admissible extension of the arguments within is potentially an admissible extension of the theory context:

Definition 4.10 *A system of arguments* $(\mathcal{A}, \rightharpoonup)$ *within an argument space* Δ *is* **reconciled** *with a theory context* $\mathcal{C}$ *if and only if:*

- *Every complete extension of* $(\mathcal{A}, \rightharpoonup)$ *given theory core* Θ *of* $\mathcal{C}$ *is a potential restriction into* Δ *of an admissible extension of* $\mathcal{C}$.
- *There exists a potential restriction* $\mathcal{E}'$ *of the accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$ *into* Δ *such that* $\mathcal{E}'$ *is an admissible extension of* $(\mathcal{A}, \rightharpoonup)$ *given* Θ.

The purpose of reconciliation is to achieve synchronisation of agent beliefs within a common argument space. If the theory context for every agent engaged in social argumentation can be shown to be reconciled with the shared system of arguments, then the theories derived from those theory contexts can be shown to be synchronised within the argument space of that system:

Theorem 4.6 *If a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *is reconciled with a set of theory contexts* $\mathcal{C}$*, then the set of all theories* Π *derived from those contexts is synchronised within the argument space* Δ *in which* $(\mathcal{A}, \rightharpoonup)$ *was generated.*

Proof 4.6 *If a shared system of arguments* $(\mathcal{A}, \rightharpoonup)$ *within an argument space* Δ *is reconciled with the theory context* $\mathcal{C}$ *of every agent* $\sigma \in \Sigma$*, where* Σ *is the set of agents engaged in an interaction* $\mathcal{I}$*, then for every theory* Π *derived from a theory context* $\mathcal{C}$ *of an agent* $\sigma \in \Sigma$*:*

- *There exists a system of arguments in* $\mathcal{C}$ *such that* Π *is derived from an admissible extension* $\mathcal{E}$ *of that system, where* $\mathcal{E}$ *is the accepted extension of* $\mathcal{C}$ *(by Definition 3.16).*
- *There exists a potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}$ *into* Δ *(by Definition 4.10).*
- *For each theory* Π_μ *held by an agent* $\mu \in \Sigma$*, it is the case that* $\mathcal{E}'$ *is a potential restriction into* Δ *of an admissible extension of* $\mathcal{C}_\mu$ *(by Definition 4.10; we know that* $\mathcal{E}'$ *is itself an admissible extension, and we know that every admissible extension in* $(\mathcal{A}, \rightharpoonup)$ *is a potential restriction into* Δ *of an admissible extension in every theory context* $\mathcal{C}_\mu$*, because* $(\mathcal{A}, \rightharpoonup)$ *is reconciled with every theory context* $\mathcal{C}_\mu$*).*

These three statements correspond to the requirements of Definition 4.8. Therefore if $(\mathcal{A}, \rightharpoonup)$ *is reconciled with every theory context* $\mathcal{C}$*, then the set of all theories* Π *derived from those theory contexts is synchronised with* Δ*.*

This then allows us to define the solution state of the distributed decision problem which we wish to solve using our portrayal mechanism. We want our mechanism to ensure that every agent in an interaction is able to reconcile their theory contexts with the portrayal for that interaction. By doing this, the beliefs of those agents will be synchronised within the argument space of the portrayal, which will (hopefully) lead to a better outcome for interaction.

4.3 Conducting Distributed Argumentation

Assume then that we have a collection of agents, each of which derives its beliefs from a defeasible theory produced by an internal argumentation process. These agents are given cause to discuss some logical problem, which defines the focus for some social argumentation process. In question then is what exactly constitutes the context for that process and how agents in dialogue can determine that context for themselves.

Whilst abstractly the critical components of social argumentation are analogous to those of internal argumentation, the underlying objective is different. Recall that in §3.2.4, we presented internal argumentation as a scientific programme like that described by [Lakatos, 1970], wherein we have a hard core and protective belt with which to determine the state of some logical theory. In social argumentation (or at least, the type of social argumentation we are interested in here), we are more interested in providing a medium through which theories can be tested and challenged, and our primary goal is to provide a means to intelligently expand the horizons of individual agents' reasoning.

As alluded to in the previous section, we are not interested in producing a single theory, but in allowing each agent involved to draw its own interpretation. Therefore for our purposes, social argumentation takes place within a distributed argumentation framework $(\mathcal{L}, \vdash, \Delta)$, generating a shared system of arguments $(\mathcal{A}, \rightharpoonup)$, which is then interpreted individually by each observing agent according to their (revised) theory contexts. The main issue then is determining the nature of the distributed framework $(\mathcal{L}, \vdash, \Delta)$. If agents are able to determine whether or not a given argument can be generated within $(\mathcal{L}, \vdash, \Delta)$, then agents can freely map arguments from their theory

contexts into Δ.

The distributed logical framework $(\mathcal{L}, \vdash)$ used for social argumentation is *potentially* the union of all frameworks used within the theory contexts of agents (i.e. if an argument is valid with respect to *any* agent's theory context, then it is valid in the distributed framework), and is *ideally* the intersection (i.e. if an argument is valid within the distributed framework, then it is valid in *all* theory contexts). If there is indeed a disparity between the logical frameworks used by agents to contribute arguments to a social argumentation process, then there will exist the possibility that some arguments will not be interpretable by all agents:

Definition 4.11 *Given a logical framework* $(\mathcal{L}, \vdash)$*, an argument* $\langle \Phi, \alpha \rangle$ *is* **invalid** *if* $(\Phi \cup \{\alpha\}) \nsubseteq \mathcal{L}$ *or* $\Phi \nvdash \alpha$*.*

If $(\mathcal{L}, \vdash)$ is the logical framework used in a theory context $\mathcal{C}$, then it can be stated that argument $\langle \Phi, \alpha \rangle$ is invalid with respect to $\mathcal{C}$.

Similarly, if a given logical framework permits an expanded notion of logical contrary (as discussed in §3.2.2), then there may exist instances where an agent claims that one argument attacks another, but there is no apparent conflict from the perspective of another peer:

Definition 4.12 *Given a logical framework* $(\mathcal{L}, \vdash)$*, an attack* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *is* **invalid** *if there does not exist a contrary relation* (α, γ) *such that* $\Psi \vdash \gamma$ *and* $\{\alpha\} \vdash \neg\gamma$*.*

If $(\mathcal{L}, \vdash)$ is the logical framework used in a theory context $\mathcal{C}$, then it can be stated that attack $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ is invalid with respect to $\mathcal{C}$.

We can assume that an agent will automatically dismiss any arguments or attacks invalid with respect to its theory context. If a distributed argumentation framework permits invalid arguments and attacks, then it will not necessarily be possible to synchronise the beliefs of agents, because certain arguments deemed admissible to some peers will be deemed inadmissible to others, and *vice versa*. It is possible to avoid making invalid arguments (and attacks) by ensuring that nothing is presumed by the logical frameworks of individual peers, but then we sacrifice the ability to avoid pedantry and make more succinct arguments (recall §3.2.2). One answer is to use dialogues which recognise where arguments are considered insufficient to peers, and allow their restatement with more explicit support [Black and Hunter, 2008]. If this is the case, then we can use the identification of invalid argument and attacks as a means to refine the logical framework under which arguments are constructed.

Given an argument $\langle \Phi, \alpha \rangle$ generated within a theory context $\mathcal{C}_\sigma$ by agent σ which is invalid with respect to another agent μ's theory context $\mathcal{C}_\mu$, there should exist an expanded argument $\langle \Phi', \alpha \rangle$ such that $\Phi' \vdash \alpha$ in any deductive framework $(\mathcal{L}, \vdash)$ (i.e. Φ is sufficient to deduce α without presuming any additional premises or rules). Thus there exist a non-empty set of sentences $S = (\Phi'/\Phi)$ such that $\emptyset \vdash_\sigma \varphi$ for all sentences $\varphi \in S$ given the logical framework $(\mathcal{L}_\sigma, \vdash_\sigma)$ of $\mathcal{C}_\sigma$. Given evidence that S cannot be presumed in a distributed argumentation framework if all peers are to interpret all arguments, an agent can effectively demote the sentences S from being subsumed by $(\mathcal{L}_\sigma, \vdash_\sigma)$ to being part of Θ of $\mathcal{C}_\sigma$ (i.e. instead of being self-evident rules of the domain, agent σ treats S as merely part of the current environment state; elements of S then appear in arguments in a shared argument system, allowing arguments to be interpreted by peers, but there is no effect on the σ's interpretation of the argument system).

In practice then, agents can produce arguments which they think to be valid, and then correct them later as necessary, using the base-line of a purely deductive logical framework as a guarantee that all arguments can be made valid eventually, provided that they lie within the argument space of the social argumentation process.

As mentioned in §3.2.2, the argument space in which argumentation occurs need not be fully-defined from the outset. Since no agent can be certain as to what arguments might be produced by peers, it is highly likely that the argument space for a given social argumentation process will adapt to the arguments already expressed — for example, in order to ensure that the argument space is sufficiently expressive as per Definition 4.9, or in order to ensure that the arguments within are balanced as per Definition 4.6. All we really need then is the ability to determine a base argument space, and then have a process by which that space can be extended on demand (and, if necessary, a process by which existing arguments are replaced by their potential expansion into the extended space). In Chapter 5, we define the argument space for an interaction portrayal as being based on producing minimal arguments supporting particular resolutions of constraints on interaction which are still sufficiently expressive enough to permit synchronisation within that argument space.

We can say then that each agent $\sigma \in \Sigma$, where Σ is the set of agents engaged in social argumentation, has a *view* $(\Theta, (\mathcal{L}, \vdash, \Delta), (\mathcal{A}_\sigma, \rightharpoonup_\sigma), \mathsf{lab})$ of a distributed social argumentation process, which is based on σ's theory context $\mathcal{C}$ and the shared system of arguments $(\mathcal{A}, \rightharpoonup)$, where:

- Θ is the theory core of $\mathcal{C}$; this theory core may be extended by the observations of other peers if σ trusts them enough to accept those observations unquestioningly.

- $(\mathcal{L}, \vdash, \Delta)$ consists of the logical framework $(\mathcal{L}, \vdash)$ of $\mathcal{C}$ and a social argument space Δ.

- $(\mathcal{A}_\sigma, \rightharpoonup_\sigma)$ is the shared system of arguments $(\mathcal{A}, \rightharpoonup)$ after the dismissal of invalid arguments and attacks.

- lab labels $(\mathcal{A}_\sigma, \rightharpoonup_\sigma)$ such that if $\mathsf{lab}(\mathbf{a}) = \mathsf{in}$ for some $\mathbf{a} \in \mathcal{A}$, then $\mathbf{a} \sqsubseteq \mathbf{b}$, where $\mathbf{b} \in \mathcal{E}$ and $\mathcal{E}$ is the accepted extension of $\mathcal{C}$.

All that remains then is for us to specify a distributed process model for the creation and maintenance interaction portrayals which will exhibit the properties defined in this chapter. The articulation of arguments into a portrayal whilst agents try to reconcile their theories with the portrayal as interaction progresses will then constitute the dialogues we envisaged at the start of this thesis, achieving our desired system.

In summary, what we have now is comprehension of how agent beliefs can be reconciled in a restricted yet focused manner by means of a common potential restriction of the beliefs of individual agents. That is, given a set of argument systems constructed under similar logical frameworks, but perhaps within very different argument spaces, each belonging to a different agent, it might be possible to construct a simpler system of arguments which adequately describes each agent's accepted conclusons within its own smaller argument space. If such a simpler system can be created, then we can state that there is some synchronicity between the beliefs of agents as they are described within each agent's personal theory context. In this light, the interaction portrayal mechanism described in the next chapter is basically a means to construct a shared system of arguments which is a potential restriction of the private beliefs agents in an interaction have about the correct outcome of interaction. Using that potential restriction, belief revision can be performed on the part of every peer such that their beliefs are synchronised within the portrayal argument space. As a consequence, the resolution of constraints on interaction (being determined by those synchronised beliefs) should be admissible to all agents.

Chapter 5

Distributed Portrayals of Interaction

In the first chapter of this thesis, we introduced the notion of an *interaction portrayal*, a device by which agents can debate possible resolutions of logical constraints imposed on an interaction by an interaction protocol using argumentation. Portrayals are intended to act as a medium through which agents can disseminate knowledge and influence one another's beliefs, in a manner akin to distributed truth maintenance. By restricting the initial scope of argumentation, and slowly expanding that scope only where necessary to adequately evaluate any conflicting opinions agents might have, we believe it to be possible to portray interactions whilst they are being enacted, as an opportunistic process.

The idea is that an interaction in motion defines a decision problem, based on the logical propositions which must be satisfied in order to bring about particular outcomes. In order to ensure that the decisions made by agents during interaction best reflect their combined wisdom, we desire that agents' beliefs are synchronised within an argument space defined by the context of the interaction. The role of the portrayal mechanism is to identify the current state of the decision problem, and then synchronise agent beliefs accordingly, 'solving' the problem until the evolution of the unfolding interaction re-defines it. The portrayal mechanism manages this by providing a framework in which agents can collectively produce a shared system of arguments, the eponymous interaction portrayal, which they then can individually reconcile with their own beliefs. If all agents are able to reconcile the portrayal with their (revised) beliefs, then those beliefs will be synchronised within the argument space explored by the portrayal, which we hypothesize will lead to objectively better outcomes for many practical interactions. Because the content of the portrayal is essentially integrated into the very belief architectures of the agents involved, the benefits of a portrayal will then

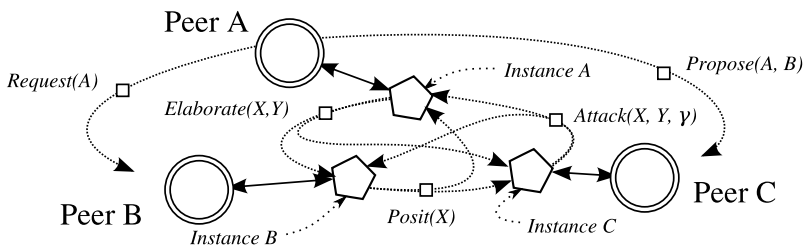
extend into later interactions.

This chapter concerns itself with the portrayal mechanism as a process for conducting distributed argumentation of the form described in Chapter 4 in order to augment interactions of the type described in Chapter 2. In §5.1 we introduce the components of a portrayal, and examine their contribution to the overall process. In §5.2 we describe how a new portrayal is created upon the initiation of a new interaction, and then in §5.3, we describe how a portrayal is defined by the unfolding interaction state. In §5.4, we describe how new arguments can be posited into a portrayal, as well as provide the operations necessary to annotate that portrayal and propagate portrayal data to individual peers. We also specify in §5.4 how agents should re-evaluate their beliefs in the presence of an updated portrayal.

5.1 Anatomy of an Interaction Portrayal

A portrayal is first created when an agent initiates a new interaction based on a chosen interaction protocol. As more agents become involved in the nascent interaction, the updated state of that portrayal is disseminated to each new peer in turn. In a distributed system, we do not assume the existence of a single shared location in which the portrayal can be stored, so instead the portrayal state is recorded independently by every agent involved in an interaction within what is referred to as a *portrayal instance*. Every such instance is kept in synchrony with every other instance of the same portrayal by means of message exchange according to some process model — these messages essentially comprise the explication dialogues referred to in the first chapter of this thesis, with the portrayal instances themselves being the accessible portion of the state of the distributed decision problem which we wish our agents to solve.

Portrayal instances are kept updated by exchanging messages whilst the greater interaction occurs around them.



Argumentation is conducted via portrayal instances whilst Peer A conducts dialogue with both Peer B and Peer C.

That decision problem can be described as follows:

1. We wish to *synchronise* (as per Definition 4.8) the decision theories Π of a set of agents Σ engaged in an interaction within an argument space Δ defined by the state of interaction such that any decision made by an agent $\sigma \in \Sigma$ during the interaction will be based on inferences at least admissible (as per Definition 3.8) to every other agent in Σ.

2. We do this by constructing a shared system of arguments $(\mathcal{A}, \rightharpoonup)$ within Δ using the same logical frameworks used by agents to derive their theories; this shared argument system is stored within the *portrayal* $\mathcal{P}$ for the given interaction. It is assumed that agents possess a common semantics for arguments, or are able to map their private arguments into such a semantics (deviation as to the precise logical frameworks used by agents is permitted however — see §5.1.4).

3. We determine Δ from the interaction state $\mathcal{S}$ of the interaction, which we assume is distributed amongst Σ as described in §2.3. From $\mathcal{S}$ we can identify a set of logical propositions Υ which must be resolved in order to determine the outcome of the interaction. The argument space of the portrayal is focused on arguing different resolutions of members of Υ and initially permits only direct assumption of claims; it is then expanded to permit attacks against elaborations of arguments in $(\mathcal{A}, \rightharpoonup)$ such that Δ is *sufficiently expressive* (as per Definition 4.9) with respect to every $\sigma \in \Sigma$'s theory Π. This ensures that arguments are kept at a minimum level of necessary detail, but also ensures that agents are always able to attack arguments if they feel justified in doing so.

4. In order to operate within an asynchronous distributed environment, we distribute the actual portrayal itself, producing for each agent $\sigma \in \Sigma$ a *portrayal instance* $\mathcal{P}[\sigma]$ which contains a copy of $(\mathcal{A}, \rightharpoonup)$ as well as additional annotations upon $(\mathcal{A}, \rightharpoonup)$ which allow σ to identify for every peer $\mu \in \Sigma$:

   - Any arguments or attacks in $(\mathcal{A}, \rightharpoonup)$ considered by μ to be invalid;
   - Any intersection between $(\mathcal{A}, \rightharpoonup)$ and μ's theory core;
   - The potential restriction into Δ of the accepted extension of μ's theory context.

5. Only once every agent $\mu \in \Sigma$ has *reconciled* (as per Definition 4.10) its theory

context $\mathcal{C}$ with $\mathcal{P}$ is agent σ then permitted to decide a constraint within interaction $\mathcal{I}$. It is expected to do so based on its (possibly revised) theory Π.

Decisions advance interaction, which may lead to the identification of new propositions to be resolved — this changes the argument space of the portrayal and so new arguments may need to be made in order to re-synchronise agent theories. This continues until the interaction is complete. This constitutes the *declarative* or definitional view of the portrayal mechanism.

5.1.1 Portraying Interaction

What this chapter provides is an algorithmic or *procedural* view of the portrayal mechanism which describes how the above definitional view can be implemented for a generic asynchronous distributed system. The basic process model can be summarised as follows:

1. Upon the initiation of a new interaction, the initiating agent σ should *portray* the interaction state $\mathcal{S}$; an initial portrayal instance $\mathcal{P}[\sigma]$ is then created which describes σ's initial arguments for or against certain portrayable propositions. The conception of new portrayal instances is specified in §5.2.

2. Being a distributed mechanism, the process model for portrayals is perhaps best understood as a collection of responses on the part of particular agents to certain events. To every such event we attribute a sub-procedure. In §5.3, we specify sub-procedures for *environmental* events — those events which define the decision problem driving a portrayal and which make use of the portrayal for practical ends:

   (a) As interaction progresses, σ will interact with other peers. Upon entering dialogue with a new peer μ, agent σ should add μ to portrayal $\mathcal{P}$ such that μ has its own instance $\mathcal{P}[\mu]$, at which point μ can posit its own arguments and react to events just as σ can (§5.3.1).

   (b) Likewise as interaction progresses, the interaction state $\mathcal{S}$ will advance such that the set of portrayable propositions Υ grows. Whenever the local interaction state $\mathcal{S}[\sigma]$ is advanced, an agent σ should check to see if new constraints have become portrayable; if new propositions are identified, then all peers should be informed so that new arguments can be posited (§5.3.2).

(c) Upon being called to resolve any constraint on interaction, an agent σ should permit all peers to finish argumentation so as to ensure that all peers have synchronised their beliefs within the argument space of portrayal $\mathcal{P}$ (§5.3.3).

Given these sub-procedures, we can use portrayals to guide interaction, provided that the individual portrayal instances used by agents are kept in synchrony.

3. Internally, argumentation is conducted by agents invoking the argue procedure (§5.4.1) to posit new arguments; arguments are then received by peers which then invoke the reconcile procedure (§5.4.11) in order to reconcile them with their own beliefs, which may then cause them to invoke argue again in response. In such a manner does argumentation continue until all agents have reconciled with the portrayal. There exist a number of operations available which are invoked either by argue, by reconcile or by one another as circumstances dictate:

   - The posit operation (§5.4.2) permits the insertion of new arguments into a portrayal $\mathcal{P}$, and is called by argue, usually in response to the addition of new peers to the interaction or to the identification of new portrayable propositions used by constraints on the interaction.
   - The elaborate operation (§5.4.3) permits the elaboration of potential arguments, usually to expose vulnerability to attack or in response to inquiries by peers.
   - The attack operation (§5.4.4) identifies where one argument attacks another; for efficiency, attack often subsumes posit and elaborate, and is often invoked in response to reconciling other agents' arguments.
   - The inquire operation (§5.4.5) is used to request elaborations from other peers; inquire can be used by agents to extend the scope of argumentation slightly where there is the promise of useful additional information.
   - The observe and unobserve operations (§5.4.6 and §5.4.7) are used by agents to identify where arguments in a portrayal contradict their theory cores (i.e. to identify arguments which must be false given available evidence). observe is usually invoked in response to the positing of new arguments by other peers; unobserve is usually invoked in response to changes in the environment changing an agent's theory core. See §5.1.5 below.

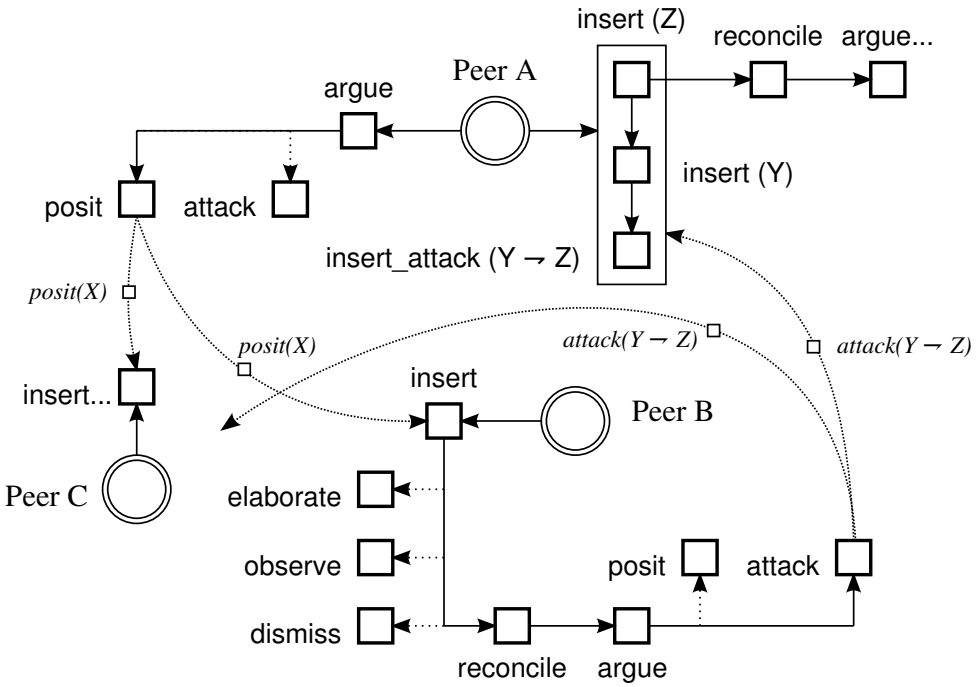


Figure 5.1: *An example of portrayal refinement as a system of procedures and operations. Peer A posits an argument (as per §5.4.1), which Peer B inserts into its portrayal instance (§5.4.2), reconciles with its beliefs (§5.4.11) and attacks with a counter-argument (§5.4.1 again); Peer C may respond similarly. Peer A will then insert any new arguments into her own portrayal instance, and make any counter-arguments deemed necessary.*

- The dismiss and expand operations (§5.4.8 and §5.4.9) are used by agents to identify and repair invalid arguments and attacks. dismiss is usually invoked in response to new arguments or attacks, whilst expand is invoked in response to dismiss, replacing an invalid argument with one more deductively explicable where possible. See §5.1.4 below.

- The accept operation (§5.4.10) is used by an agent to declare the extension of a portrayal which it currently chooses to accept as a potential restriction of its own beliefs; accept is usually invoked after an invocation of reconcile if an agent has found need to revise its beliefs. See 5.1.6 below.

These operations all produce messages which are sent to all peers in an interaction, allowing them to keep their portrayal instances updated and ensuring that the beliefs of agents are synchronised as per Definition 4.8.

4. At the end of its role in interaction, an agent should report the closure of that role. Once all agents have done so, the portrayal can be disposed of. In the meantime, all agents can continue to contribute to the portrayal should they be inclined to do so, and all agents will receive portrayal updates. This is described in §5.3.4 just after the other environmental events.

5. The computational complexity of the portrayal mechanism is subservient to the complexity of extracting new arguments from an agent's theory context. Thus the efficacy of the portrayal mechanism is dependent on the extent to which agents are already aware of the arguments supporting or conflicting with their beliefs. In summary:

    - The bulk of computation is in the private argumentation processes used by agents to define their theory contexts. If an agent goes into interaction with a fully-formed and interpreted system of arguments, then the complexity of constructing portrayals will be limited to the cost of deriving potential arguments from internal arguments. Conversely, if an agent goes into interaction with just a set of candidate assumptions and constructs new arguments *in situ*, then the primary source of complexity will be the construction and initial evaluation of arguments.
    - It should be noted however that if agents are truly autonomous, then they must surely perform a certain amount of inference prior to interaction in order to determine the need for interaction and select a protocol which will likely result in a desired outcome. This means that each agent can be expected to enter interaction having already enacted an internal argumentation process (or equivalent defeasible reasoning process) in order to formulate expectations for constraints on interaction. In this respect, a significant portion of possible argumentation will necessarily have been performed prior to constructing an interaction portrayal, leaving only argument comparison and recombination.
    - In the worst case, agents will need to fully elaborate upon their claims, expanding the argument space of the portrayal to encompass that of all agents' theory contexts. In the average case however, significant computation can be saved by focusing on the minimal amount of argumentation necessarily to ensure synchronisation of agent beliefs, depending on the characteristics of the domain in which argumentation is conducted.

The portrayal mechanism is decidable given a finite system of arguments in the theory context of the executing agent; depending on the logic used, the argumentation process used within an agent's theory context might not be. Agents must keep tight control over the argument spaces in which they generate arguments so as to ensure tractability, in precisely the same way that any agent in a complex environment must maintain control of its reasoning processes in order to remain responsive..

The operations and procedures described in the remainder of this chapter can be shown to ensure the properties described in the previous chapter. First however, it is necessary to make clear exactly what a portrayal, or more precisely a portrayal *instance* contains.

5.1.2 Specification of a Portrayal Instance

A *portrayal instance* is a copy of an interaction portrayal belonging to a specific agent. For any distributed implementation of the portrayal mechanism, a portrayal is synonymous with its instances — from the perspective of any particular agent, a reference to an interaction's portrayal is equivalent to a reference to its own instance of that portrayal:[1]

Definition 5.1 *A* **portrayal instance** $\mathcal{P}[\sigma]$ *used by an agent* σ *with a theory context* $\mathcal{C}$ *to describe the state of a portrayal* $\mathcal{P}$ *of an interaction* $\mathcal{I}$[2] *from the perspective of* σ *can be described by a tuple* $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *where:*

- Σ *is the set of identifiers for agents involved in* $\mathcal{I}$ *and which therefore also hold instances of* $\mathcal{P}$*. For every peer* $\mu \in \Sigma$*, it is assumed that there exists a theory context* $\mathcal{C}_\mu$ *which agent* μ *will use to generate and interpret arguments in* $\mathcal{P}$*.*

- Υ *is the set of* **portrayable propositions** *in* $\mathcal{I}$ *identified by agents in* Σ *such that if* $\varphi \in \Upsilon$*, then* φ *is evaluated as part of a constraint imposed on* $\mathcal{I}$ *(see §5.2.1 later).*

- $(\mathcal{A}, \rightharpoonup)$ *is the system of arguments described by* $\mathcal{P}$ *such that every argument* $\mathbf{a} \in \mathcal{A}$ *and every attack* $\mathbf{a} \rightharpoonup \mathbf{b}$ *is valid within a theory context* $\mathcal{C}_\mu$ *of an agent* $\mu \in \Sigma$*.*

[1] The distinction between a portrayal and a portrayal instance is that an interaction conceptually has one portrayal whilst it may in practice have many portrayal instances which individually describe different views of the portrayal.

[2] As per Definition 2.1.

- *The* **invalid argument** *function* $\mathsf{inv}_{\mathcal{A}} : \Sigma \to 2^{\mathcal{A}}$ *is a function mapping each agent* $\mu \in \Sigma$ *to a set of arguments* $\mathsf{inv}_{\mathcal{A}}(\mu) \subseteq \mathcal{A}$ *such that if* $\mathbf{a} \in \mathsf{inv}_{\mathcal{A}}(\mu)$*, then* $\mathbf{a}$ *is an invalid argument with respect to* $\mathcal{C}_\mu$ *as per Definition 4.11.*

- *The* **invalid attack** *function* $\mathsf{inv}_{\rightharpoonup} : \Sigma \to 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ *is a function mapping each agent* $\mu \in \Sigma$ *to a set of argument pairs* $\mathsf{inv}_{\rightharpoonup}(\mu)$ *such that if* $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\mu)$*, then* $\mathbf{a} \rightharpoonup \mathbf{b}$ *according to* $(\mathcal{A}, \rightharpoonup)$*, but* $\mathbf{a} \rightharpoonup \mathbf{b}$ *is an invalid attack with respect to* $\mathcal{C}_\mu$ *as per Definition 4.12.*

- *The* **observation** *function* $\mathsf{obs} : \Sigma \to 2^{\Gamma}$ *is a function mapping each agent* $\mu \in \Sigma$ *to a set of sentences* $\mathsf{obs}(\mu) \subseteq \Gamma$ *(where* Γ *is the set of all sentences* φ *such that there exists an argument* $\langle \Phi, \alpha \rangle \in \mathcal{A}$ *in which* $\Phi \vdash \neg\varphi$ *according to* $\mathcal{C}_\mu$*) such that if* $\varphi \in \mathsf{obs}(\mu)$*, then* $\Theta \vdash \varphi$ *according to* $\mathcal{C}_\mu$*, where* Θ *is the theory core of* $\mathcal{C}_\mu$*.*

- *The* **acceptance** *function* $\mathsf{acc} : \Sigma \to 2^{\mathcal{A}}$ *is a function mapping each agent* $\mu \in \Sigma$ *to a set of arguments* $\mathsf{acc}(\mu) \subseteq \mathcal{A}$ *such that if* $\mathbf{a} \in \mathsf{acc}(\mu)$*, then* $\mathbf{b} \in \mathcal{E}$*, where* $\mathcal{E}$ *is the accepted extension of* $\mathcal{C}_\mu$*, and* $\mathbf{a} \sqsubseteq \mathbf{b}$ *according to* $\mathcal{C}_\mu$*.*

A portrayal instance records the arguments and attacks observed by an agent along with the set of *portrayable propositions*, which is the set of logical propositions identified thus far in an interaction as being used in constraints on that interaction and which therefore need to be discussed by peers. A portrayal instance also records any arguments or attacks declared invalid by peers (§5.1.4), any relevant propositions which a peer has claimed to have observed directly (§5.1.5), as well as each peer's declared accepted extension of the argument system within the portrayal (§5.1.6).

For brevity, if an argument $\mathbf{a} \in \mathcal{A}$ for the set of arguments $\mathcal{A}$ of a portrayal instance $\mathcal{P}[\sigma]$ of a portrayal $\mathcal{P}$, then $\mathbf{a} \in \mathcal{P}$. Similarly, if $\mathbf{a} \rightharpoonup \mathbf{b}$ according to $\mathcal{P}[\sigma]$, then it can be said that $\mathbf{a} \rightharpoonup \mathbf{b}$ according to $\mathcal{P}$.

From the theory context $\mathcal{C}$ of an agent σ and its portrayal instance $\mathcal{P}[\sigma]$, we can infer σ's view of the distributed context in which the portrayal $\mathcal{P}$ is constructed. In this case, we have a view $(\Theta, (\mathcal{L}, \vdash, \Delta), (\mathcal{A}, \rightharpoonup), \mathsf{lab})$ where:

- Θ is a consistent subset of $\bigcup_{\mu \in \Sigma} \mathsf{obs}(\mu)$ as described in §5.1.5 below.

- $(\mathcal{L}, \vdash, \Delta)$ consists of the logical framework $(\mathcal{L}, \vdash)$ of $\mathcal{C}$ and the argument space Δ of $\mathcal{P}$ as described in §5.1.3 below.

- $(\mathcal{A}, \rightharpoonup)$ is the system of arguments in $\mathcal{P}[\sigma]$ after dismissing every argument in

$\mathsf{inv}_{\mathcal{A}}(\sigma)$ and every attack $\mathbf{a} \rightharpoonup \mathbf{b}$ referenced by $\mathsf{inv}_{\rightharpoonup}(\sigma)$, and is a potential restriction into Δ of the system of arguments in $\mathcal{C}$.

- lab labels $(\mathcal{A}, \rightharpoonup)$ such that if $\mathsf{lab}(\mathbf{a}) = \mathsf{in}$ for some $\mathbf{a} \in \mathcal{A}$, then $\mathbf{a} \sqsubseteq \mathbf{b}$,[3] where $\mathbf{b} \in \mathcal{E}$ and $\mathcal{E}$ is the accepted extension of $\mathcal{C}$ (i.e. the labelling of $(\mathcal{A}, \rightharpoonup)$ accepts the potential restriction of σ's beliefs into the portrayal).

Thus the content of a portrayal instance $\mathcal{P}[\sigma]$ is sufficient to describe the state of a social argumentation process from the perspective of agent σ provided that it is possible to infer from it the argument space Δ of $\mathcal{P}$. Fortunately, because we are able to model the portrayal mechanism in such a way as to specify how and when arguments are entered into a portrayal, it is possible to reconstruct the argument space of a portrayal simply from the system of arguments already within it and the set of portrayable propositions.

5.1.3 Computing the Argument Space of a Portrayal

The argument space of a portrayal (which we will sometimes refer to simply as the *portrayal space*) is determined over the course of interaction by the constraints imposed on it by an interaction's protocol, the arguments already in the portrayal and the potential attacks that the agents involved in an interaction can make. This final criterion ensures that the argument space of a portrayal is sufficiently expressive, but also makes it practically impossible for any single peer in an interaction to infer the true portrayal space for itself. Fortunately, this is not a problem, because it is not necessary for an agent to know the complete argument space in order for it to determine whether a given argument drawn from its beliefs is within that space.

Whether or not an argument can be inserted into a portrayal can be determined by an agent using its portrayal instance and its theory context:

Definition 5.2 *An argument $\langle \Phi, \alpha \rangle$ is* **within** *the argument space Δ of a portrayal $\mathcal{P}$ with respect to a context $\mathcal{C}$ (i.e. $\langle \Phi, \alpha \rangle \in \Delta$ from the perspective of an agent with theory context $\mathcal{C}$) if and only if:*

- *Either $\alpha \in \Upsilon$, $\neg\alpha \in \Upsilon$ (where Υ is the set of portrayable propositions in $\mathcal{P}$) or there exists an argument $\mathbf{a}' \in \mathcal{P}$ such that $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$, where $\mathbf{a} \in \mathcal{C}$ and $\mathbf{a}' \sqsubseteq \mathbf{a}$.*

[3] Recall from Chapter 3 that $\mathbf{a} \sqsubseteq \mathbf{b}$ states that argument $\mathbf{a}$ is potentially argument $\mathbf{b}$ if elaborated upon.

- *There does not exist an elaboration* $\mathbf{b} \in \mathcal{E}_A$*, where* $\mathcal{E}_A$ *is the accepted extension of* $\mathcal{C}$*, such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{b}$ *and* $\mathbf{c} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{c} \in \mathcal{P}$*, but* $\mathbf{c} \not\rightharpoonup \langle \Phi, \alpha \rangle$*, unless there also exists an elaboration* $\mathbf{d} \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{d} \not\sqsubseteq \mathbf{b}$ *and* $\mathbf{c} \not\rightharpoonup \mathbf{d}$*.*

- *There does not exist a potential argument* $\mathbf{e}$ *such that:* $\mathbf{e} \sqsubset \langle \Phi, \alpha \rangle$*, but* $\mathbf{e} \not\sqsubseteq \mathbf{f}$ *for some argument* $\mathbf{f} \in \mathcal{P}$*; and* $\mathbf{e} \in \Delta$*.*

An argument is within the portrayal space from the perspective of a given agent if it fulfils three criteria: the argument attempts to resolve a portrayable proposition or else attacks another argument already in the portrayal; the argument is not potentially another argument in the portrayal (thus conferring no new information); and the argument is otherwise expressed as simply as possible whilst still acknowledging any necessary attacks from arguments already in the portrayal (i.e. if, in the agent's own belief, an argument requires a particular supporting proposition to be true, and that proposition is already contradicted by another argument within the portrayal, then the agent must make that conflict explicit even if it believes the other argument to be inadmissible; otherwise it can just keep the argument as simple as possible until it needs to defend it from other attacks).

A portrayal space is effectively self-expanding — it always permits new attacks on and alternative elaborations of existing arguments, albeit perhaps without much detail initially. This ensures that the portrayal space always becomes expressive enough to sufficiently describe the practical differences between agent theories, in accordance with Definition 4.9 (which in turn ensures that synchronisation of beliefs within the portrayal space is feasible):

Theorem 5.1 *The portrayal space* Δ *of a portrayal* $\mathcal{P}$ *is sufficiently expressive (as per Definition 4.9) provided that all agents posit any attacks dictated by their beliefs.*

Proof 5.1 *The argument space* Δ *of a portrayal* $\mathcal{P}$ *is sufficiently expressive with respect to a theory* Π *generated within a theory context* $\mathcal{C}$ *if and only if:*

- *There exists a system of arguments* $(\mathcal{A}, \rightharpoonup)$ *such that* Π *can be derived from an admissible extension* $\mathcal{E}$ *of* $(\mathcal{A}, \rightharpoonup)$ *(provided by* $\mathcal{C}$*).*

- *There exists a potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}$ *into* Δ *(always true, though* $\mathcal{E}'$ *may be empty).*

- *If there exists an argument* $\mathbf{a} \in \Delta$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}'$*, then provided that there exists an argument* $\mathbf{c} \in \mathcal{E}$ *such that* $\mathbf{c} \rightharpoonup \mathbf{a}$*, there exists an argument* $\mathbf{d} \in \Delta$ *such that* $\mathbf{d} \sqsubseteq \mathbf{c}$ *and* $\mathbf{d} \rightharpoonup \mathbf{a}$*.*

Assume that there exists an argument $\mathbf{a} \in \Delta$ *as described above and there likewise exists an argument* $\mathbf{c} \in \mathcal{E}$ *such that* $\mathbf{c} \rightharpoonup \mathbf{a}$*. We know that there exists an argument* $\langle \Phi, \alpha \rangle$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{c}$ *and* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$*, because:*

- *If* $\mathbf{c} \rightharpoonup \mathbf{a}$*, then* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$ *(by Theorem 4.1).*
- $\mathbf{c} \sqsubseteq \mathbf{c}$ *(by Definition 4.1).*

We shall assume however that for all possible $\langle \Phi, \alpha \rangle$*, it is the case that* $\langle \Phi, \alpha \rangle \notin \Delta$*. This means, by Definition 5.2, that there is no argument* $\langle \Phi, \alpha \rangle$ *for which:*

- *Either* $\alpha \in \Upsilon$*,* $\neg\alpha \in \Upsilon$ *(where* Υ *is the set of portrayable propositions in* $\mathcal{P}$*) or there exists an argument* $\mathbf{e}' \in \mathcal{P}$ *such that* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{e}$*, where* $\mathbf{e} \in \mathcal{C}$ *and* $\mathbf{e}' \sqsubseteq \mathbf{e}$*.*
- *There does not exist an elaboration* $\mathbf{f} \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{f}$ *and* $\mathbf{g} \rightharpoonup \mathbf{f}$ *for some argument* $\mathbf{g} \in \mathcal{P}$*, but* $\mathbf{g} \not\rightharpoonup \langle \Phi, \alpha \rangle$*, unless there also exists an elaboration* $\mathbf{h} \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{h} \not\sqsubseteq \mathbf{f}$ *and* $\mathbf{g} \not\rightharpoonup \mathbf{h}$*.*
- *There does not exist a potential argument* $\mathbf{i}$ *such that:* $\mathbf{i} \sqsubset \langle \Phi, \alpha \rangle$*, but* $\mathbf{i} \not\sqsubseteq \mathbf{j}$ *for some argument* $\mathbf{j} \in \mathcal{P}$*; and* $\mathbf{i} \in \Delta$*.*

It can be seen however that if $\mathbf{a} \in \mathcal{P}$*, then* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$*, fulfilling the first criterion under that condition. As for the second criterion:*

- *If there exists an elaboration* $\mathbf{f} \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{f}$ *and* $\mathbf{g} \rightharpoonup \mathbf{f}$ *for some argument* $\mathbf{g} \in \mathcal{P}$*, but* $\mathbf{g} \not\rightharpoonup \langle \Phi, \alpha \rangle$*, then there exists an argument* $\mathbf{k}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{k} \sqsubseteq \mathbf{f}$*, but* $\mathbf{g} \rightharpoonup \mathbf{k}$*.*
- *If* $\mathbf{f} \sqsubseteq \mathbf{c}$*, then* $\mathbf{k} \sqsubseteq \mathbf{c}$ *and* $\mathbf{k} \rightharpoonup \mathbf{a}$*, and therefore* $\mathbf{k}$ *can replace* $\langle \Phi, \alpha \rangle$*, thus fulfilling the criterion.*
- *If* $\mathbf{c} \sqsubset \mathbf{f}$*, then* $\mathbf{f}$ *can replace* $\mathbf{c}$*; given that* $\mathbf{k} \sqsubseteq \mathbf{f}$ *and* $\mathbf{k} \rightharpoonup \mathbf{a}$*, argument* $\mathbf{k}$ *can replace* $\langle \Phi, \alpha \rangle$*, thus fulfilling the criterion.*
- *Otherwise,* $\mathbf{c}$ *is a separate elaboration of* $\langle \Phi, \alpha \rangle$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{c} \not\sqsubseteq \mathbf{f}$*, in which case there exists an argument* $\mathbf{l}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{l} \sqsubseteq \mathbf{c}$ *and* $\mathbf{l} \not\sqsubseteq \mathbf{f}$*. Therefore* $\mathbf{l} \sqsubseteq \mathbf{c}$ *and* $\mathbf{l} \rightharpoonup \mathbf{a}$*, so* $\mathbf{l}$ *can replace* $\langle \Phi, \alpha \rangle$*, thus fulfilling the criterion.*

For the third criterion, we simply need to observe that if there is *a potential argument* $\mathbf{i} \in \Delta$ *such that* $\mathbf{i} \sqsubset \langle \Phi, \alpha \rangle$ *and* $\mathbf{i} \not\sqsubseteq \mathbf{j}$ *for some argument* $\mathbf{j}$ *in* $\mathcal{P}$*, then* $\mathbf{i}$ *can replace* $\langle \Phi, \alpha \rangle$*, because* $\mathbf{i}$ *fulfils all criteria. Fulfilment of all three criteria contradicts the assumption that there exists no argument* $\langle \Phi, \alpha \rangle \in \Delta$ *for which* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{c}$ *and* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{a}$*.*

Therefore the argument space Δ of a portrayal $\mathcal{P}$ becomes sufficiently expressive with respect to a theory Π generated within a theory context $\mathcal{C}$ upon the insertion of any argument $\mathbf{a} \in \Delta$ such that $\mathbf{a} \rightharpoonup \mathbf{b}$ for some argument $\mathbf{b} \in \mathcal{E}'$, where $\mathcal{E}'$ is the potential restriction into Δ of the accepted extension $\mathcal{E}$ of $\mathcal{C}$.

If agents use different logical frameworks (i.e. assume different logical sentences to be common knowledge, and thus unnecessary for expression within arguments), then an agent might insert into a portrayal arguments and attacks which are invalid to its peers. The portrayal mechanism provides agents with the ability to report any invalid arguments and attacks, as well as rectify them (see §5.1.4), but the asynchronous nature of a distributed system permits the generation of further argumentation in the meantime (i.e. whilst one agent declares an argument invalid, other agents are concurrently elaborating upon and attacking that argument). Because however invalid arguments are still part of the portrayal, the argument space still permits attacks against them, thus allowing the above theorem to still hold. This has no undue effect on the interpretation of the portrayal by agents which consider certain arguments to be invalid however, because they simply ignore those arguments upon evaluation.

5.1.4 Dismissing Invalid Arguments and Attacks

As discussed in §3.2.2, it is possible to simplify the system of arguments generated within an argumentation framework by allowing certain axioms and 'undeniable' rules to be integrated into the logical framework used to construct arguments, allowing agents to omit those axioms and rules from arguments, as well as allowing arguments to be attacked based on notions of mutual exclusion other than explicit negation of a proposition. It is of course not necessarily the case that such axioms and rules are common knowledge to every agent engaged in an interaction. It may be the case then that certain arguments and attacks will be considered by peers to be invalid as per Definitions 4.11 and 4.12.

The basic purpose of the invalid argument function $\mathsf{inv}_{\mathcal{A}}$ and the invalid attack function $\mathsf{inv}_{\rightharpoonup}$ of a portrayal instance $\mathcal{P}[\sigma]$ is to record which arguments and attacks have been declared invalid by an agent during the life-time of the portrayal; this gives peers an opportunity to posit more expansive arguments, rather than allowing them to assume that current arguments adequately support their claims. The rejection of an argument usually indicates a presumption not held by peers, whilst the rejection of an attack usually indicates that at least one peer does not recognise that two sentences are

mutually exclusive. Either case signifies that an agent needs to spell out its reasoning more explicitly.

Example 5.1 *Let us consider the following two arguments regarding agent Alanna's trustworthiness.*

$$\begin{aligned}\mathbf{a} = \langle\ \{\ & \mathsf{collaborated}(\mathsf{alanna},\mathsf{dante},\mathsf{laboratory}) \\ & \mathsf{analogous}(\mathsf{laboratory},\mathsf{library}) \\ & \forall X.\ \neg\mathsf{incident}(\mathsf{laboratory},X), \\ & \forall W,X,Y,Z.\ \mathsf{collaborated}(W,X,Y) \wedge \mathsf{analogous}(Y,Z) \\ & \qquad \rightarrow (\exists A.\ \mathsf{incident}(Y,A)) \vee \mathsf{trustworthy}(X,Z)\ \} \\ & \forall X.\ \mathsf{trustworthy}(\mathsf{alanna},X)\ \rangle\end{aligned}$$

$$\mathbf{b} = \langle\ \{\ \mathsf{leaked}(\mathsf{laboratory},\mathsf{data})\ \},\ \mathsf{incident}(\mathsf{laboratory},\mathsf{leak}(\mathsf{data}))\ \rangle$$

Argument **a** *is posited by Dante, and argument* **b** *constitutes Charlotte's response. The intuition here is that the leaking of data constitutes an incident occurring during collaboration between Alanna and Dante, which undermines the assertion that there were no incidents during that collaboration. This intuition may not be immediately apparent to all agents however. Let us assume that Dante dismisses argument* **b** *such that* $\mathbf{b} \in \mathsf{inv}_{\mathcal{A}}(\mathsf{dante})$. *In this instance, Charlotte can re-factor her attack such that it more explicitly spells out her point:*

$$\begin{aligned}\mathbf{b}' = \langle\ \{\ & \mathsf{leaked}(\mathsf{laboratory},\mathsf{data}), \\ & \forall X,Y.\ \mathsf{leaked}(X,Y) \rightarrow \mathsf{incident}(X,\mathsf{leak}(Y))\ \}, \\ & \mathsf{incident}(\mathsf{laboratory},\mathsf{leak}(\mathsf{data}))\ \rangle\end{aligned}$$

Dante must then concede that argument $\mathbf{b}'$ *is valid, and respond accordingly. It may be worth noting that* $\mathbf{b}'$ *is more open to attack than argument* **b** *— for example, an argument could conceivably attack the assertion that a leak constitutes an incident.*

It is evident that the closer an agent adheres to basic deduction, without making any presumptions about the ability of their peers to understand any implicit reasoning, the more likely that the agent will not generate any invalid arguments and attacks. On the other hand, the more complex the domain in which argumentation occurs, the more burdensome it is to generate complete deductive arguments. Ultimately, the domain in which argumentation occurs and the sophistication of the agents within it will determine the level of cleverness which can be presumed in the logical frameworks used by agents. Concerning invalid arguments:

- The validity of an argument $\mathbf{a} \in \mathcal{P}$ is evaluated by an agent σ with respect to its theory context $\mathcal{C}$ upon its insertion into portrayal instance $\mathcal{P}[\sigma]$ (see the insert

function, §5.4.2); an invalid argument is declared as such by invocation of the dismiss operation (§5.4.8).

- An agent can replace an invalid argument **a** by invocation of the expand operation (§5.4.9) in response to receiving a $\mathsf{dismiss}(\mathbf{a})$ message from a peer.

- An agent will automatically remove an argument **a** from $\mathsf{inv}_{\mathcal{A}}(\sigma)$ for all peers σ if it is replaced by an elaboration **b** in $\mathcal{P}$ (even if **b** is itself invalid — **b** will be declared as such automatically, replacing **a**; see §5.4.2).

- A peer may attempt to elaborate upon an argument $\mathbf{a} \in \mathcal{P}$, replacing it with an argument **b** which is invalid with respect to the theory context of agent σ; if **a** is valid, then σ will not consider it to be potentially **b** (by Definition 4.1, the support for a valid argument cannot be wholly derived from an invalid argument, otherwise it would be valid itself) in which case it will treat **b** as a separate argument within $\mathcal{P}[\sigma]$. This has no undue effect on argumentation as **b** will be dismissed prior to interpretation, but will allow σ the continued use of argument **a**. If **b** is later replaced with a valid argument **c**, then **a** will automatically be replaced by **c** as well (evident in §5.4.2).

Concerning invalid attacks:

- The validity of an attack $\mathbf{a} \rightharpoonup \mathbf{b}$ according to $\mathcal{P}$ is evaluated by an agent σ with respect to its theory context $\mathcal{C}$ upon its identification in portrayal instance $\mathcal{P}[\sigma]$ (see the insert_attack function, §5.4.4), and is declared by invocation of the dismiss operation (§5.4.8).

- An agent σ can replace the attacking argument **a** of an invalid attack pair $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\sigma)$ by invocation of the expand operation (§5.4.9) in response to receiving a $\mathsf{dismiss}(\mathbf{a} \rightharpoonup \mathbf{b})$ message from a peer.

- Elaborations of an argument automatically inherit the attack relations of that argument (as justified by Theorem 4.1); if either the attacking argument **a** or the defending argument **b** of an invalid attack pair $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\sigma)$ is elaborated upon, then the elaboration replaces the elaborated upon argument in $\mathsf{inv}_{\rightharpoonup}(\sigma)$ (evident in §5.4.2 again).

Assuming that agents are able to replace any invalid arguments or attacks with ones which are valid to all agents (for example by using the expand operation to replace

arguments with ones which more explicitly describe any deduction made), it can be expected that all invalid arguments and attacks will be purged by the end of interaction.

5.1.5 Asserting Observations into the Portrayal

Agents are permitted to annotate a portrayal by asserting that certain (pertinent) sentences are already known by it to be true or false, regardless of the arguments surrounding them. This advocacy of certain propositions informs an agent's peers that the advocate has no current intention of accepting any interpretation of the portrayal argument system which contradicts those propositions. An agent should declare any observation which can be deduced from the theory core of its beliefs which is directly contradicted by any argument in the portrayal — any such contradictory arguments can then be formally dismissed as described in §3.2.3.

Ideally, only sentences which are *known* to be true by an agent should be asserted as such, where 'known' entails that a sentence can be inferred by sound deduction from axiomatic terms or direct observation (including introspection of an agent's own state). If this is universally the case, then the agents in an interaction can always simply take the union of all sets $\mathsf{obs}(\sigma)$ for all agents $\sigma \in \Sigma$ of a portrayal $\mathcal{P}$ to produce an internally consistent common theory core. This core can then be used to find all commonly-admissible argument extensions.

Example 5.2 *Consider the following system of arguments, which might be part of a portrayal:*

$$\begin{aligned}
\mathbf{a} = \langle\ \{\ & \mathsf{permission}(\mathsf{charlotte}, \mathsf{access}(\mathsf{alanna}, \mathsf{library})),\\
& \mathsf{controller}(\mathsf{charlotte}, \mathsf{library}),\\
& \forall X, Y, Z.\ \mathsf{permission}(X, \mathsf{access}(Y, Z)) \wedge \mathsf{controller}(X, Z) \rightarrow \mathsf{access}(Y, Z)\ \},\\
& \mathsf{access}(\mathsf{alanna}, \mathsf{library})\ \rangle
\end{aligned}$$

$$\begin{aligned}
\mathbf{b} = \langle\ \{\ & \mathsf{controller}(\mathsf{dante}, \mathsf{library}),\\
& \exists! X.\ \mathsf{controller}(X, \mathsf{library})\ \},\\
& \neg\mathsf{controller}(\mathsf{charlotte}, \mathsf{library})\ \rangle
\end{aligned}$$

$$\begin{aligned}
\mathbf{c} = \langle\ \{\ & \neg\mathsf{patron}(\mathsf{alanna}, \mathsf{library}),\\
& \forall X, Y.\ \mathsf{access}(X, Y) \rightarrow \mathsf{patron}(X, Y)\ \},\\
& \neg\mathsf{access}(\mathsf{alanna}, \mathsf{library})\ \rangle
\end{aligned}$$

$$\begin{aligned}
\mathbf{d} = \langle\ \{\ & \mathsf{access}(\mathsf{dante}, \mathsf{library}),\\
& \neg\mathsf{patron}(\mathsf{dante}, \mathsf{library})\ \},\\
& \exists X.\ \mathsf{access}(X, \mathsf{library})) \wedge \neg\mathsf{patron}(X, \mathsf{library})\ \rangle
\end{aligned}$$

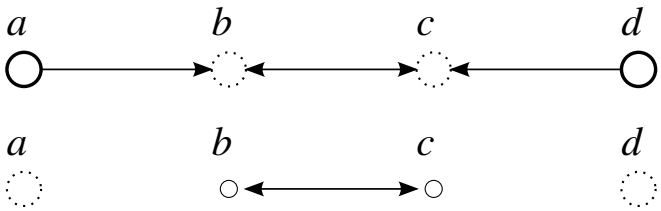


Figure 5.2: *The grounded extension of the system of arguments described in Example 5.2 before and after observations are applied.*

By default, arguments **b** *and* **d** *are admissible, whilst arguments* **a** *and* **c** *are not. If however the following observations are made:*

$$\begin{aligned} \mathsf{obs}(\mathsf{charlotte}) &= \{\ \mathsf{controller}(\mathsf{charlotte}, \mathsf{library})\ \} \\ \mathsf{obs}(\mathsf{dante}) &= \{\ \neg\mathsf{access}(\mathsf{dante}, \mathsf{library})\ \} \end{aligned}$$

We can immediately determine arguments **b** *and* **d** *to be unacceptable (because* **b***'s claim is directly refuted and* **d** *has been undermined) leaving only arguments* **a** *and* **c** *to choose between. If either argument can be undercut by an undefeated argument, or if further observations clarify the domain, then the argument system will provide an unambiguous interpretation.*

Complex systems rarely admit ideal circumstances however. Even assuming the competency of agents to determine that a given sentence is 'known' rather than simply believed true, only a small number of propositions can actually be determined to be certain in a dynamic and unpredictable environment. In practice, it is more useful for an agent to advocate things which it can confidently infer as true under reasonable conditions based on the improbability that they might be otherwise. Such an approach would permit such intuitions as 'knowing' that the sky remains blue or that one's house has not been moved over to the next valley along, even when out of sight and therefore strictly speaking not provable given the full range of possibility. No matter how sensibly conducted however, any abduction risks being shown to be incorrect, and so a balance must be struck between what is or is not subject to argument (this could easily be contextual — one could imagine a recursive series of defeasible processes where the theory core of one process is the product of another, more fundamental one). Given then that agents are independent, the unavoidable conclusion is that the union of all sets $\mathsf{obs}(\sigma)$ for all agents $\sigma \in \Sigma$ of a portrayal $\mathcal{P}$ might not be jointly consistent after all.

Recall however that portrayals are simply vessels for introducing new concepts and percepts to the hypothesis spaces of agents, and that ultimately each agent interprets

that information in accordance with its private defeasible reasoning mechanism. Whilst it would be foolish to ignore the observations of one's peers, should those observations be inconsistent, one can always be selective. For example, one peer may be considered more qualified to make observations in a given context than another. Should there be no clear preference order to selecting one percept over another however, it may be best simply to treat contradictory elements as normal assumptions or claims and disregard the advocacy of agents in favour of the topology of the argument system. Concerning observations:

- An agent σ determines whether the theory core Θ of its theory context $\mathcal{C}$ justifies the addition of a new observation φ into $\mathsf{obs}(\sigma)$ whenever a new argument $\mathbf{a} \in \mathcal{P}$ is inserted into portrayal instance $\mathcal{P}[\sigma]$ (see insert function, §5.4.2); if so, σ invokes the observe operation (§5.4.6).

- If an agent believes that an argument $\mathbf{a} \in \mathcal{P}$ should be dismissed, but the pertinent observation φ is not explicitly contradicted by **a** (i.e. for all elaborations $\mathbf{b} \in \mathcal{E}$, where $\mathcal{E}$ is the accepted extension of theory context $\mathcal{C}$, such that $\mathbf{a} \sqsubset \mathbf{b}$, it is the case that whilst $\neg\varphi$ cannot be derived from **a**, but it can also be seen that $\neg\varphi$ *can* be derived from **b**), then the agent can elaborate upon **a** specifically in order to then observe φ in order to dismiss the elaboration (see §5.4.3 and §5.4.6).

- If the theory core Θ of an agent σ's theory context $\mathcal{C}$ changes, then σ should invoke function observe or function unobserve as necessary to update $\mathsf{obs}(\sigma)$; agent σ can invoke observe to add new assertions to $\mathsf{obs}(\sigma)$ (§5.4.6) or invoke unobserve to remove assertions from $\mathsf{obs}(\sigma)$ (§5.4.7).

- When interpreting the system of arguments in theory context $\mathcal{C}$, an agent σ can use the results of obs as a preference ordering on admissible extensions of $\mathcal{C}$, such that arguments supported by peer observations are favoured over arguments which are not. Agent σ can, subject to its own sense of caution, add assertions in obs to theory core Θ, but only if σ trusts the peer which made the observation and the observation does not make Θ internally inconsistent. The precise mechanism to do this is particular to the implementation of agent σ.

Ultimately, the basis on which an agent chooses to add peers' observations to its theory core, or alternatively uses observations for some kind of preference ordering of arguments, is at the discretion of the individual agents. An agent's peers can infer how an

agent σ treats their observations by examining $\mathsf{acc}(\sigma)$ within their individual portrayal instances.

5.1.6 Accepting Extensions of Portrayal Arguments

The purpose of argumentation is to determine which assumptions are reasonable given the evidence available and which conclusions then follow. Portrayals exist to allow argumentation to be performed in restricted circumstances, getting the best results possible in such conditions. Based on the system of arguments in a portrayal and given the other factors already discussed, each agent in an interaction is expected to select the argument extension which best aligns with its (possibly revised) beliefs. There is no requirement that all agents select the *same* extension — indeed, it is not permissible given the presumed autonomy of individual agents to be otherwise. Ideally, the combined insights and arguments of all agents would leave only one admissible preferred extension of the argument system within the portrayal. In practice, the best we can aim for is that all agents will make decisions which, whilst perhaps not aligned with the beliefs of all peers, will be admissible to every peer nevertheless. At worst, disagreements about the environment state (in particular 'known' facts) will lead to agents making decisions which are entirely unacceptable to one or more peers. Even in this instance however, the portrayal will provide insight into why such a scenario had come to pass, which could feed into the agent's future decisions (whether to find conclusive evidence forestalling some irreconcilable difference or to avoid interacting with certain agents within certain domains).

The purpose of the acceptance function acc of a portrayal is to inform agents of the standing assumptions of their peers; in particular, whether certain arguments have been defeated or not from the perspective of those peers. An agent can tell from the acceptance of particular arguments whether a given peer is behaving sceptically or credulously, or whether the peer has produced a complete interpretation of a given argument system or just part of it.

Example 5.3 *Consider the following system of arguments embedded in a portrayal $\mathcal{P}$:*

$$\begin{aligned}\mathbf{a} = \langle\ \{\ & \mathsf{researcher}(\mathsf{alanna},\mathsf{astronomy}),\\ & \mathsf{domain}(\mathsf{library},\mathsf{astronomy}),\\ & \forall X,Y,Z.\ \mathsf{researcher}(X,Y)\wedge\mathsf{domain}(Z,Y)\rightarrow\mathsf{eligible}(X,Z)\ \},\\ & \mathsf{eligible}(\mathsf{alanna},\mathsf{library})\ \rangle\end{aligned}$$

$$\begin{aligned}\mathbf{b} = \langle\ \{\ & \neg\exists X.\ \mathsf{published}(\mathsf{alanna},X),\\ & \forall X,Y.\mathsf{researcher}(X,Y)\rightarrow\mathsf{published}(X,Y)\ \},\end{aligned}$$

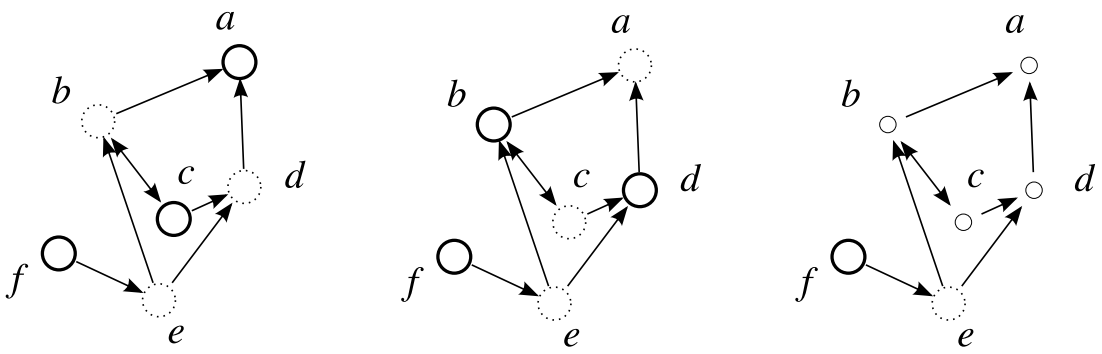


Figure 5.3: *A system of arguments, interpreted in three different ways by Alanna, Benjamin and Charlotte.*

$\neg\exists X.\ \mathsf{researcher}(\mathsf{alanna}, X)\ \rangle$

$\mathbf{c} = \langle\ \{\ \mathsf{researcher}(\mathsf{alanna}, \mathsf{astronomy}),$
$\forall X, Y.\mathsf{researcher}(X, Y) \rightarrow \mathsf{published}(X, Y)\ \},$
$\mathsf{published}(\mathsf{alanna}, \mathsf{astronomy})\ \rangle$

$\mathbf{d} = \langle\ \{\ \neg\mathsf{published}(\mathsf{alanna}, \mathsf{astronomy}),$
$\forall X, Y.\ \mathsf{expert}(X, Y) \rightarrow \mathsf{published}(X, Y)\ \},$
$\neg\mathsf{expert}(\mathsf{alanna}, \mathsf{astronomy})\ \rangle$

$\mathbf{e} = \langle\ \{\ \mathsf{publication}(\mathsf{alanna}, \mathsf{research}),$
$\mathsf{subject}(\mathsf{research}, \mathsf{astronomy}),$
$\forall X, Y, Z.\ \mathsf{publication}(X, Y) \wedge \mathsf{subject}(Y, Z) \rightarrow \mathsf{published}(X, Z)\ \},$
$\mathsf{published}(\mathsf{alanna}, \mathsf{astronomy}\ \rangle$

$\mathbf{f} = \langle\ \{\ \mathsf{subject}(\mathsf{research}, \mathsf{climatology})\ \},\ \neg\mathsf{subject}(\mathsf{research}, \mathsf{astronomy})\ \rangle$

Now assume that the acceptance function acc *of a portrayal instance* $\mathcal{P}[\sigma]$ *describes the following:*

$\mathsf{acc}(\mathit{alanna}) = \{\ \mathbf{a}, \mathbf{c}, \mathbf{f}\ \}$
$\mathsf{acc}(\mathit{benjamin}) = \{\ \mathbf{b}, \mathbf{d}, \mathbf{f}\ \}$
$\mathsf{acc}(\mathit{charlotte}) = \{\ \mathbf{f}\ \}$

Alanna and Benjamin accept opposing (but admissible) interpretations of the argument system, whilst Charlotte is maintaining a sceptical (grounded) view. If Alanna (or anybody else) wishes to persuade both Benjamin and Charlotte that Alanna is eligible for access to the library (the claim of argument **a***), she will need to conclusively defeat arguments* **b** *and* **d**.

Concerning argument acceptance:

- The set of arguments $\mathsf{acc}(\sigma)$ accepted by an agent σ is the potential restriction into Δ of portrayal $\mathcal{P}$ of the accepted extension $\mathcal{E}$ of $\mathcal{C}$, where $\mathcal{C}$ is the theory context of σ.

- If an agent σ posits a new argument **a** into $\mathcal{P}$, whether on its own or as an attacker, then **a** is added to $\mathsf{acc}(\sigma)$ automatically, as σ can only posit arguments which are potentially arguments in $\mathcal{E}$ of $\mathcal{C}$.

- An agent σ reasserts $\mathsf{acc}(\sigma)$ entirely by invoking operation accept; this is done after σ reconciles $\mathcal{C}$ with $\mathcal{P}$ if $\mathsf{acc}(\sigma)$ is no longer a potential restriction into Δ of $\mathcal{E}$ (see §5.4.10).

- It is possible for there to be an argument $\mathbf{a} \in \mathsf{acc}(\sigma)$ where $\mathbf{a} \notin \mathcal{P}$. This is because elaborations of arguments do *not* inherit acceptance from their potential arguments. This is because there may be many elaborations of the same potential argument, not all of which are accepted in the theory context $\mathcal{C}$ of a given peer σ; it is also possible that **a** has been elaborated into an invalid argument with respect to $\mathcal{C}$ as described in §5.1.4.

Having over-viewed the essential components of a portrayal, we can now concentrate on how a portrayal is generated and updated.

5.2 Initialising an Interaction Portrayal

The portrayal mechanism first comes into play upon the initiation of a new interaction. An agent initiates a new interaction by selecting an interaction protocol and adopting an initial role and accompanying process model. Upon instantiating that role, the agent is then able to construct a new portrayal instance which will form the basis of a new interaction portrayal for the nascent interaction. *Portrayal conception* is the act of creating a new portrayal based on a newly-adopted process model and an agent's starting beliefs.

Definition 5.3 *An initial interaction state $\mathcal{S}$ is portrayed by an agent σ by first identifying the portrayable propositions Υ already present within the interaction model used by $\mathcal{S}$ and then conceiving a new portrayal instance $\mathcal{P}$ focused on Υ:*

$$\mathsf{portray}(\mathcal{S},\sigma) \quad \leftrightarrow \quad \left(\begin{array}{l} \mathsf{context}(\sigma,\mathcal{C}) \ \wedge \ \mathsf{portrayable\_propositions}(\mathcal{S},\Upsilon) \ \wedge \\ \mathsf{conceive}(\mathcal{C},\Upsilon,\mathcal{P}[\sigma]) \ \wedge \ \mathsf{assert}(\sigma,\mathcal{P}[\sigma]) \end{array} \right)$$

Where:

- $\mathsf{context}(\sigma, \mathcal{C})$ *is true if* $\mathcal{C}$ *is the theory context of agent* σ.
- $\mathsf{portrayable\_propositions}(\mathcal{S}, \Upsilon)$ *is true if* Υ *is the set of portrayable propositions described in interaction state* $\mathcal{S}$ *(see §5.2.1 below).*
- $\mathsf{conceive}(\mathcal{C}, \Upsilon, \mathcal{P}[\sigma])$ *is true if* $\mathcal{P}[\sigma]$ *is the portrayal instance describing the initial arguments supporting or debunking instances of the portrayable propositions* Υ *which can be derived from theory context* $\mathcal{C}$ *(see 5.2.2 below).*
- $\mathsf{assert}(\sigma, \mathcal{P}[\sigma])$ *attributes the portrayal instance* $\mathcal{P}[\sigma]$ *to agent* σ*; it can then be retrieved by invoking* $\mathsf{portrayal}(\sigma, \mathcal{P}[\sigma])$.

Formally, portrayal conception is invoked upon generation of an initial interaction state $\mathcal{S}$ after selection of a protocol $\mathbb{P}$ by agent σ. We can thus invoke portray during initial interaction state selection as described in §2.3.1, modifying Definition 2.11 as shown below:

Definition 5.4 *An interaction state* $\mathcal{S}$ *models a system state* $\mathbb{S}$ *if* $\mathcal{S}$ *is the initial state of an interaction based on a protocol* $\mathbb{P}$ *selected by an agent* σ *which it considers to be applicable to* $\mathbb{S}$*:*

$$\mathsf{models}(\mathbb{S}, \mathcal{S}) \quad \leftarrow \quad \mathsf{selection}(\sigma, \mathbb{S}, \mathbb{P}) \wedge \mathsf{initial\_state}(\mathbb{P}, \mathcal{S}) \wedge \mathsf{portray}(\mathcal{S}, \sigma)$$

Where all is as described in Definition 2.11 except:

- $\mathsf{portray}(\mathcal{S}, \sigma)$ *is true if agent* σ *is able to create an initial portrayal instance from interaction state* $\mathcal{S}$.

Note that the portrayal instance is not handed over to the process model for interaction. Instead the portrayal is developed in parallel with interaction and is referred to by a modified interaction process as defined in §5.3.

5.2.1 Identifying Portrayable Propositions

It is necessary, if a group of agents are to discuss valid resolutions for constraints placed upon interaction, that those agents are able to identify the logical propositions which those constraints depend on prior to their actual resolution. The ability of any given agent to do this is dependent however on the protocol for interaction and the interaction state at various points during interaction. In particular, a simple harvesting of propositions used in constraints in a given protocol is not desirable, primarily for two reasons:

- It may not be yet known whether a given constraint will apply in this particular instance of interaction.
- The propositions within a constraint may not be sufficiently instantiated to allow for intelligent discussion.

The first reason should be fairly self-explanatory — portraying unused propositions will invite arguments into a portrayal which have no bearing on the interaction at hand, and if so permitted, the computational cost of using a portrayal will be harder to justify. The second reason is also important however. Consider a proposition "X trusts Y", where neither X nor Y is bound to any value. It is likely that the intention of such a proposition in a constraint within a protocol is to allow agents to evaluate whether a particular X trusts a particular Y, or to select a Y given a particular X, or possibly even to select an X given a particular Y. In any of those cases, there is some restriction on satisfactory instances of the proposition and thus the worthwhile arguments that should be produced within a portrayal. Less likely is that the intention is to find an arbitrary pairing of X and Y which satisfies the proposition — in such a case, a portrayal would have to permit arguments for and against *any* such arbitrary pairing, which could be significant in number to say the least.

In order to ensure that arguments are only produced for intended resolutions of a given constraint, the preferred policy for portrayable propositions is that any proposition is portrayable only when instantiated to the greatest extent to which it can be instantiated prior to actual resolution. For example, if we have a proposition "X trusts Y" with the intention to select a Y given a specific X, then we wait until X is instantiated before discussing Y. We can formalise this notion as follows:

Definition 5.5 *The set of portrayable propositions Υ described by an interaction state $\mathcal{S}$ is the union of portrayable propositions in every role model adopted in $\mathcal{S}$*[4]*:*

$$\mathsf{portrayable\_propositions}(\mathcal{S},\Upsilon) \quad \leftrightarrow \quad \mathsf{role\_models}(\mathcal{S},\mathcal{M}) \wedge \left(\Upsilon = \bigcup_{\mathcal{M}\,[R,\sigma]\in\mathcal{M}} \Upsilon_\sigma \;\middle|\; \mathsf{portrayables}(\mathcal{M}\,[R,\sigma],\Upsilon_\sigma) \right)$$

Where:

- $\mathsf{role\_models}(\mathcal{S},\mathcal{M})$ *is true if $\mathcal{M}$ is the set of all process models $\mathcal{M}\,[R,\sigma]$ adopted by agents in the interaction described by interaction state $\mathcal{S}$ such that $\mathcal{M}\,[R,\sigma]$ describes role R as adopted by agent σ.*

[4] Role models were defined back in §2.3.2.2.

- $\mathsf{portrayables}(\mathcal{M}\,[R,\sigma],\Upsilon)$ *is true if* Υ *is the set of portrayable propositions in process model* $\mathcal{M}\,[R,\sigma]$.

Proposition portrayable_propositions works by extracting the set of process models for roles adopted by agents so far in the interaction described by interaction state $\mathcal{S}$ and within every such model, finding the set of propositions portrayable.

Definition 5.6 *Trivially, the set of portrayable propositions in a given role clause* $\mathcal{M}\,[R,\sigma]$ *is the subset of propositions in* $\mathcal{M}\,[R,\sigma]$ *which have reached a portrayable state:*

$$\mathsf{portrayables}(\mathcal{M}\,[R,\sigma],\Upsilon) \quad \leftrightarrow \quad \mathsf{propositions}(\mathcal{M}\,[R,\sigma],\Gamma) \wedge \left(\Upsilon = \bigcup_{\varphi\in\Gamma} \varphi \mid \mathsf{portrayable}(\mathcal{M}\,[R,\sigma],\varphi)\right)$$

Where:

- $\mathsf{propositions}(\mathcal{M}\,[R,\sigma],\Gamma)$ *is true if* Γ *is the set of logical propositions referred to in role clause* $\mathcal{M}\,[R,\sigma]$.

- $\mathsf{portrayable}(\mathcal{M}\,[R,\sigma],\varphi)$ *is true if proposition* φ *is portrayable given the state of role clause* $\mathcal{M}\,[R,\sigma]$.

Proposition portrayables finds all propositions which might be portrayable in a given role model, and then checks every such proposition separately against that model. A proposition is portrayable if there is no intersection between the unbound variables within the proposition, and any unbound variables otherwise found in the model that would be encountered prior to the resolution of proposition in any execution of the model (i.e. all unbound variables within the proposition are introduced by the proposition and thus can be expected to be bound by satisfying the proposition, rather than by satisfying an earlier proposition or by receiving a message from another peer).

Definition 5.7 *A proposition* φ *is portrayable if the set of unbound variables in* φ *and the set of unbound variables found prior to* φ *in role clause* $\mathcal{M}\,[R,\sigma]$ *is disjoint:*

$$\mathsf{portrayable}(\mathcal{M}\,[R,\sigma],\varphi) \quad \leftrightarrow \quad \left(\begin{array}{ll} \mathsf{unbound\_variables}(\{\varphi\},V_\varphi) & \wedge \\ \mathsf{prior\_variables}(\mathcal{M}\,[R,\sigma],\varphi,V_{\mathcal{M}}\,[R,\sigma]) & \wedge \\ V_\varphi \cap V_{\mathcal{M}}\,[R,\sigma] = \emptyset & \end{array}\right)$$

Where:

- unbound_variables(Γ, V) *is true if* V *is the set of unbound variables referred to within propositions in set* Γ.

- prior_variables$(\mathcal{M}\ [R, \sigma], \varphi, V)$ *is true if* V *is the set of unbound variables referred to within role clause* $\mathcal{M}\ [R, \sigma]$ *prior to proposition* φ.

Particular implementations of the portrayal mechanism, protocol specifications or interaction states may slightly influence the results of invoking certain propositional functions (e.g.propositions or prior_variables), especially if termination is to be guaranteed. For example, in LCC, one might expect that an agent may only analyse constraints in the models for specific roles adopted, rather than in sub-roles which have not been formally assumed yet, even if the assumption of such roles is inevitable. Ultimately, all that is necessary is that propositions are successfully identified as portrayable before they are resolved — earlier identification rather than later identification is nice, but not vital.

Example 5.4 *Recall briefly the example of distributed interaction described over the course of Chapter 2. In particular, recall the adoption of the* acquire_access *protocol of Chapter 1 as described in Example 2.7. Alanna had assumed the role of* applicant *in order to acquire access to* library*:*

a(applicant(library), alanna) ::
request $\Rightarrow$ a(advocate(library), *Advocate*)
$\leftarrow \neg$ accessible(alanna, library) $\wedge$ patron(*Advocate*, library) then ...

From the outset, the propositions $\neg$ accessible(alanna, library) *and* patron(*Advocate*, library) *are portrayable; the former is fully instantiated, whilst the latter provides the first reference to* Advocate*, from which we can infer that it is intended for the advocate agent to be determined by this constraint, rather than by a (non-existent) earlier one.*

Conversely, the proposition trusts(*Controller*, *Advocate*) *in the* controller *role model is* not *portrayable:*

a(controller(*Resource*), *Controller*) ::
recommend(*Applicant*) $\Leftarrow$ a(advocate(*Resource*), *Advocate*) then
(permit(*Applicant*, access(*Resource*))
$\leftarrow$ ... $\wedge$ trusts(*Controller*, *Advocate*) $\wedge$ ...).

This is because neither variable term has been instantiated and both terms are expected *to be instantiated prior to resolution of the constraint (*Controller *will be instantiated upon an agent adopting the* controller *role, and* Advocate *will be instantiated*

for the controller upon reception of the recommend *message from the agent in the* advocate *role). In this case we want both variables instantiated before portrayal, both to prevent the proliferation of pointless instances and because if the advocate agent rejects Alanna's plea, then the constraint will never be tested anyway.*

For a new interaction, only the process model for the initial role adopted by the initiating agent will be analysed, and only propositions found in initial constraints (immediately prior to the first action taken by the agent in its new role) will likely qualify as being portrayable.

5.2.2 Portrayal Conception

The conceive operation constructs a new portrayal instance and generates initial arguments within a minimal argument space.

conceive($\mathcal{C}$, Υ, $\mathcal{P}[\sigma]$) — *An agent* σ *with a theory context* $\mathcal{C}$ *conceives a new portrayal instance* $\mathcal{P}$ *focused on a set of portrayable propositions* Υ *if and only if:*

- *Agent* σ *has initiated a new interaction* $\mathcal{I}$ *adhering to a protocol* $\mathbb{P}$.

Assuming that this condition has been met, a new portrayal instance $\mathcal{P}[\sigma]$ *is defined such that* $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$*, where:*

1. $\Sigma = \{\sigma\}$.
2. *For each argument* $\mathbf{a} \in \mathcal{U}$*, where* $\mathcal{U}$ *is the unrejected extension of* $\mathcal{C}$*, if there exists a potential argument* $\mathbf{b} \sqsubseteq \mathbf{a}$ *such that* $\mathbf{b} \in \Delta$*, then* $\mathbf{b} \in \mathcal{A}$ *of* $\mathcal{P}[\sigma]$.
3. *Whilst there exist any arguments* $\mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{b} \notin \Delta$ *of* $\mathcal{P}$*, replace* $\mathbf{b}$ *in* $\mathcal{A}$ *with* $\mathbf{c}$*, where* $\mathbf{c} \in \Delta$ *and* $\mathbf{b} \sqsubset \mathbf{c} \sqsubseteq \mathbf{a}$ *given an argument* $\mathbf{a} \in \mathcal{U}$ *(i.e. if any conflicting arguments are put into the portrayal, ensure that they are sufficiently elaborated upon to illustrate the conflicts between them).*
4. *If there exist two arguments* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$ *such that* $\mathbf{c} \rightharpoonup \mathbf{d}$ *according to* $\mathcal{C}$*, and* $\mathbf{a} \sqsubseteq \mathbf{c}$ *and* $\mathbf{b} \sqsubseteq \mathbf{d}$*, then* $\mathbf{a} \rightharpoonup \mathbf{b}$ *according to* $\mathcal{P}$.
5. $\mathsf{inv}_{\mathcal{A}}(\sigma) = \emptyset$.
6. $\mathsf{inv}_{\rightharpoonup}(\sigma) = \emptyset$.
7. $\mathsf{obs}(\sigma) = \emptyset$.
8. $\mathbf{a} \in \mathsf{acc}(\sigma)$ *if and only if* $\mathbf{a} \in \mathcal{P}$ *and there exists an argument* $\mathbf{b} \in \mathcal{E}$*, where* $\mathcal{E}$ *is the accepted extension of* $\mathcal{C}$*, such that* $\mathbf{a} \sqsubseteq \mathbf{b}$.

Portrayal conception draws upon arguments from the *unrejected* extension of a theory context — that is, the arguments which an agent has not ruled out entirely. The reason we consider unrejected rather than merely accepted arguments is because this affords the opportunity for an agent's peers to clarify for it whether or not undecided arguments should be accepted or rejected. Note that the more extensive an agent's interpretation of its own theory context, the less significant the difference between the accepted and unrejected extension (recall the discussion on argument labelling in §3.2.1). Thus, a credulous agent will likely posit only what it has already accepted to be true prior to additional evidence, whilst a sceptical agent will be more likely to posit only possibilities, without committing to accepting any of them.

Example 5.5 *Let us assume then that Alanna has initiated interaction based on the* acquire_access *protocol, and has identified that the propositions* $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ *and* $\mathsf{patron}(X,\mathsf{library})$ *are portrayable. Let us assume that Alanna can extract the following accepted arguments from her theory context:*

$$\begin{aligned}
\mathbf{a}_1 &= \langle\ \{\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \},\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \rangle\\
\mathbf{b}_1 &= \langle\ \{\ \mathsf{access}(\mathsf{benjamin},\mathsf{library}),\\
&\qquad \forall X,Y.\ \mathsf{access}(X,Y)\rightarrow\mathsf{patron}(X,Y)\ \},\\
&\quad\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \rangle
\end{aligned}$$

Alanna can also extract the following unrejected arguments (she's not sure about Dante):

$$\begin{aligned}
\mathbf{c}_1 &= \langle\ \{\ \mathsf{access}(\mathsf{dante},\mathsf{library}),\\
&\qquad \forall X,Y.\ \mathsf{access}(X,Y)\rightarrow\mathsf{patron}(X,Y)\ \},\\
&\quad\ \mathsf{patron}(\mathsf{dante},\mathsf{library})\ \rangle\\
\mathbf{d}_1 &= \langle\ \{\ \mathsf{banned}(\mathsf{dante},\mathsf{library}),\\
&\qquad \forall X,Y.\ \mathsf{patron}(X,Y)\wedge\mathsf{banned}(X,Y)\rightarrow\mathsf{false},\\
&\quad\ \neg\mathsf{patron}(\mathsf{dante},\mathsf{library})\ \rangle
\end{aligned}$$

All of these arguments can be mapped into portrayal $\mathcal{P}$*:*

$$\begin{aligned}
\mathbf{a}' &= \langle\ \{\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \},\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \rangle\\
\mathbf{b}' &= \langle\ \{\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \},\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \rangle\\
\mathbf{c}' &= \langle\ \{\ \mathsf{patron}(\mathsf{dante},\mathsf{library})\ \},\ \mathsf{patron}(\mathsf{dante},\mathsf{library})\ \rangle\\
\mathbf{d}' &= \langle\ \{\ \neg\mathsf{patron}(\mathsf{dante},\mathsf{library})\ \},\ \neg\mathsf{patron}(\mathsf{dante},\mathsf{library})\ \rangle
\end{aligned}$$

Alanna will only elaborate upon her arguments if need arises. Thus we have an initial portrayal instance $\mathcal{P}[\mathsf{alanna}]$*, where:*

- $\Sigma = \{\mathsf{alanna}\}$.
- $\Upsilon = \{\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library}),\mathsf{patron}(X,\mathsf{library})\}$.

- $\mathcal{A} = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}'\}$, *such that* $\mathbf{c}' \rightharpoonup \mathbf{d}'$ *and* $\mathbf{d}' \rightharpoonup \mathbf{c}'$.
- $\mathsf{inv}_{\mathcal{A}}(\mathsf{alanna}) = \mathsf{inv}_{\rightharpoonup}(\mathsf{alanna}) = \mathsf{obs}(\mathsf{alanna}) = \emptyset$.
- $\mathsf{acc}(\mathsf{alanna}) = \{\mathbf{a}', \mathbf{b}'\}$.

Given that no other instances of the new portrayal yet exist, there is no need to communicate any new information to peers. Thus initialising a portrayal is very simple — an agent simply asserts the resolutions it expects for sufficiently instantiated constraints imposed on interaction. What we need to consider now is what happens to the portrayal as the interaction develops.

5.3 Responding to Changes in the Interaction State

In order to function correctly in an asynchronous environment, the portrayal for an interaction is considered to be an entity which exists parallel to the interaction, developing at its own pace in response to the arguments, counter-arguments and observations made by the peers involved with it. Of course to be useful, there must be points at which an interaction process and an interaction portrayal actually interact with each other. Aside from portrayal conception at the outset of a new interaction, there are four notable points of intersection between portrayal and interaction:

The addition of new peers — Whenever a new agent is involved with the interaction, that agent should be able to contribute to the interaction portrayal.

Advancement of local interaction state — Whenever an agent makes progress in an interaction, such that a message is dispatched or received, a new role is adopted, or some action deemed significant is performed, there is a possibility that the set of portrayable propositions in an interaction changes.

Constraint resolution — The primary purpose of an interaction portrayal is to actually help agents resolve constraints imposed upon interaction by its protocol 'better'. Thus any active lines of argument need to be resolved immediately prior to attempting to satisfy an interaction constraint.

The end of interaction — Finally, interactions come to an end. Rather than have an agent cease contributing to a portrayal at the end of its own role in interaction

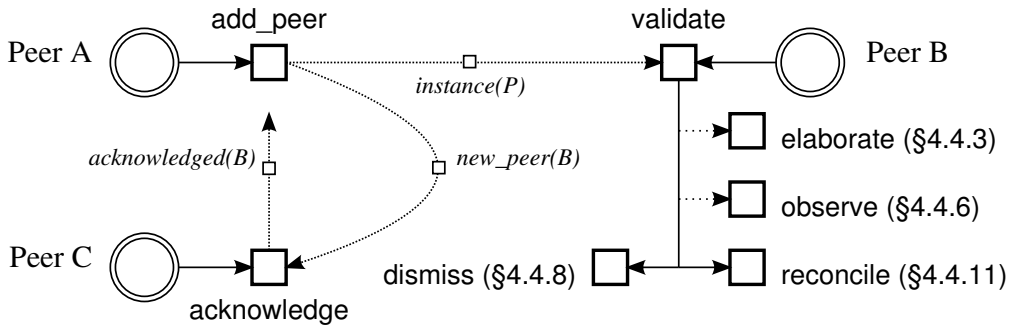


Figure 5.4: *Upon being inducted into an interaction by peer A, peer B must validate the portrayal instance it has been given whilst existing peers (such as peer C) update their portrayal instances to account for B's presence.*

however, it would be preferable if agents were able to continue making and absorbing arguments until the entire distributed interaction comes to a close should they so desire.

Once we know how an interaction portrayal interacts with the interaction process for a distributed dialogue based on an interaction protocol, we can then concentrate on the independent development of the portrayal itself.

5.3.1 Adding Peers to the Interaction

Obviously there needs to be more than one agent involved with the development of an interaction portrayal for there to be any discussion within that portrayal. An agent is attributed a new portrayal instance when it is first inducted into the interaction to which a portrayal is attached. In other words, an agent is attributed a portrayal instance when it first receives a message from any other agent acting out a role in a given interaction.

There must therefore be, in the execution model for distributed interaction, a point at which the model is able to identify a new agent to induct into the interaction — this point will be at the first dispatch of a message to the new agent. We can demonstrate when a new copy of a portrayal's argument system is transferred to a peer within the rewrite rules defined in Definition 2.15 for LCC role clauses — it is a simple matter to then extrapolate equivalent circumstances in other protocol languages:

Definition 5.8 *Upon dispatching a message M in role R to an agent σ_r in role R_r, an agent σ should designate the act of message dispatch as closed within its process*

model $\mathcal{M}\,[R,\sigma]$*:*

$$M \Rightarrow \mathsf{a}(R_r,\sigma_r) \quad \xrightarrow{R,M_i,M_i,\mathcal{S}\,[\sigma],\{\mathsf{m}(\mathsf{a}(R,\sigma),\mathsf{a}(R_r,\sigma_r),M)\}} \quad \mathsf{c}(M \Rightarrow \mathsf{a}(R_r,\sigma_r))$$
$$\textit{if} \quad \mathsf{portrayal}(\sigma,\mathcal{P}\,[\sigma]) \;\wedge\; \begin{pmatrix} \mathsf{member}(\sigma_r,\mathcal{P}\,[\sigma]) & \vee \\ \mathsf{add\_peer}(\mathcal{P}\,[\sigma],\sigma_r) & \end{pmatrix}$$

This rewrite rule replaces the rewrite rule for message dispatch in Definition 2.15. It adds two alternate conditions, either of which must be satisfied:

- $\mathsf{portrayal}(\sigma,\mathcal{P}\,[\sigma])$ *is true if* $\mathcal{P}\,[\sigma]$ *is the instance of portrayal* $\mathcal{P}$ *held by agent* σ*.*
- $\mathsf{member}(\sigma,\mathcal{P}\,[\sigma])$ *is true if* $\sigma \in \Sigma$*, where* Σ *is the set of agents with instances of portrayal* $\mathcal{P}$*.*
- $\mathsf{add\_peer}(\mathcal{P}\,[\sigma],\sigma)$ *confers a copy of portrayal instance* $\mathcal{P}\,[\sigma]$ *to a peer* σ *and informs all peers* $\mu \in \Sigma$ *that* σ *is to be added to the set of agents involved in portrayal* $\mathcal{P}$*.*

By this means can we ensure that every agent inducted into an interaction has a copy of the portrayal and is able to contribute to it.

Operation add_peer adds a new agent to the portrayal by conferring a copy of a portrayal instance to the given agent and updating the set of agents known to have portrayal instances in every other portrayal instance known to exist:

$\mathsf{add\_peer}(\mathcal{P}\,[\sigma],\mu)$ *— An agent* σ *gives peer* μ *a copy of its portrayal instance* $\mathcal{P}\,[\sigma]$ *and informs all other peers* $\mu \in \Sigma$ *of* $\mathcal{P}\,[\sigma]$ *that* μ *should be added to* Σ *if and only if:*

- *Agent* σ *is the sender and agent* μ *is the recipient of at least one message* $M \in \mathcal{I}$*, where* $\mathcal{I}$ *is the interaction dialogue to which portrayal* $\mathcal{P}$ *is attached.*
- $\mathcal{P}\,[\sigma] = (\Sigma,\Upsilon,(\mathcal{A},\rightharpoonup),\mathsf{inv}_{\mathcal{A}},\mathsf{inv}_{\rightharpoonup},\mathsf{obs},\mathsf{acc})$ *and* $\mu \notin \Sigma$*.*

A message $\mathsf{instance}(\mathcal{P}\,[\mu])$ *is then dispatched to agent* μ*, where:*

- $\mathcal{P}\,[\mu] = (\Sigma \cup \{\mu\},\Upsilon,(\mathcal{A},\rightharpoonup),\mathsf{inv}_{\mathcal{A}},\mathsf{inv}_{\rightharpoonup},\mathsf{obs},\mathsf{acc})$*.*

A message $\mathsf{new\_peer}(\mu)$ *is also dispatched to all agents in* Σ*. Moreover, until* σ *receives a message* $\mathsf{acknowledged}(\mu)$ *from every agent in* Σ*, agent* σ *will relay a copy of any (other) message received to* μ*.*

In order to ensure that a new peer does not miss any important arguments or other declarations between being inducted into interaction and being made known to all other

peers in the interaction, the agent responsible for conferring a portrayal instance to the peer ensures that it receives a copy of every portrayal-oriented message the agent receives itself.

Upon reception of a message $\mathsf{instance}(\mathcal{P}[\sigma])$, an agent σ should assert $\mathcal{P}[\sigma]$ as its own and attempt to reconcile it with its own theory context:

Definition 5.9 *An agent σ acquires a portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message* $\mathsf{instance}(\mathcal{P}[\sigma])$*:*

$$\mathsf{assert}(\sigma, \mathcal{P}[\sigma]) \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{instance}(\mathcal{P}[\sigma])) & \wedge \\ \neg \exists X.\ \mathsf{portrayal}(\sigma, X) & \wedge \\ \mathsf{validate}(\mathcal{P}[\sigma]) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M)$ *is true if agent σ has received a message M from one of its peers in Σ of $\mathcal{P}$.*
- $\mathsf{portrayal}(\sigma, \mathcal{P}[\sigma])$ *is true if $\mathcal{P}[\sigma]$ is the portrayal instance of $\mathcal{P}$ held by agent σ.*
- $\mathsf{validate}(\mathcal{P}[\sigma])$ *attempts to reconcile the theory context $\mathcal{C}$ of agent σ with portrayal $\mathcal{P}$ such that $\mathcal{C}$ and $\mathcal{P}$ fulfil Definition 4.10.*

The validate predicate checks whether or not any argument or attack in a new portrayal instance should be dismissed either due to observation or due to being invalid:

$\mathsf{validate}(\mathcal{P}[\sigma])$ — *Given a new portrayal instance $\mathcal{P}[\sigma]$ by means other than portrayal conception, an agent σ with theory context $\mathcal{C}$ should validate the arguments and attacks within $\mathcal{P}$ provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *and* $\sigma_s \in \Sigma$.

If this condition is met, then:

1. *For each argument $\langle \Phi, \alpha \rangle \in \mathcal{A}$, if an elaboration $\langle \Psi, \alpha \rangle \in \mathcal{C}$ of $\langle \Phi, \alpha \rangle$ is dismissed as per Definition 3.15 because of a sentence φ such that $\Theta \vdash \varphi$ and $\Psi \vdash \neg\varphi$ (where Θ is the theory core of $\mathcal{C}$), and there exists no alternative elaboration $\mathbf{a} \in \mathcal{U}$ (where $\mathcal{U}$ is the unrejected extension of $\mathcal{C}$), such that $\langle \Phi, \alpha \rangle \sqsubset \mathbf{a}$:*
   - *There must exist an argument $\langle \Psi', \alpha \rangle$ such that $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi', \alpha \rangle \sqsubseteq \langle \Psi, \alpha \rangle$ and $\Psi' \vdash \neg\varphi$, and for which there is no alternative argument $\langle \Psi'', \alpha \rangle$ such that $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi'', \alpha \rangle \sqsubset \langle \Psi', \alpha \rangle$ and $\Psi'' \vdash \neg\varphi$.*

- *If* $\langle \Psi', \alpha \rangle = \langle \Phi, \alpha \rangle$*, then invoke* observe$(\mathcal{P}[\sigma]', \{\varphi\})$.
- *Otherwise, invoke* elaborate$(\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle, \langle \Psi', \alpha \rangle)$ *before then invoking* observe$(\mathcal{P}[\sigma]', \{\varphi\})$.

2. *For each argument* $\mathbf{a} \in \mathcal{A}$*, if* $\mathbf{a}$ *is invalid with respect to* $\mathcal{C}$ *as per Definition 4.11, then invoke* dismiss$(\mathcal{P}[\sigma]', \mathbf{a})$.
3. *For every attack relation* $\mathbf{a} \rightharpoonup \mathbf{b}$ *according to* $\mathcal{P}$*, if* $\mathbf{a} \rightharpoonup \mathbf{b}$ *is invalid with respect to* $\mathcal{C}$*, then invoke* dismiss$(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b})$.
4. *Invoke* reconcile$(\mathcal{P}[\sigma], \mathcal{A})$.

The above predicate will invoke observe (§5.4.6) if it believes an argument can be dismissed by observation, elaborating upon that argument if necessary to illustrate its belief. It does this by finding the simplest argument which is still evidently dismissable given an observed percept; if that simplest argument is subsumed by the argument already within the portrayal, then the argument itself is left untouched. Otherwise, an elaborate operation is invoked (see §5.4.3). It will also invoke dismiss (§5.4.8) for any argument or attack which it believes to be invalid. Finally, it invokes reconcile. The reconcile function is specified in §5.4.11; it ensures that any arguments in a portrayal which are new to a given agent are absorbed into the agent's theory context and it then motivates the articulation of any arguments the agent can make which fit into the portrayal argument space.

Upon reception of a message new_peer(μ), an agent σ with a portrayal instance $\mathcal{P}[\sigma]$ should update $\mathcal{P}[\sigma]$ and acknowledge μ:

Definition 5.10 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* new_peer(μ)*:*

$$\text{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \text{received}(\sigma, \text{new\_peer}(\mu), \sigma_s) & \wedge \\ \text{portrayal}(\sigma, \mathcal{P}[\sigma]) & \wedge \\ \mathcal{P}[\sigma]' = \text{acknowledge}(\mathcal{P}[\sigma], \mu, \sigma_s) & \end{array} \right)$$

Where:

- received(σ, M, σ_s) *is true if agent* σ *has received a message* M *from* σ_s *in* Σ *of* $\mathcal{P}$.
- acknowledge$(\mathcal{P}[\sigma], \mu, \sigma_s)$ *returns an updated portrayal instance* $\mathcal{P}[\sigma]$ *such that agent* μ *is within the set of agents* Σ *of* $\mathcal{P}[\sigma]$.

Function acknowledge ensures that an agent is known to be involved with a portrayal and is acknowledged as such to the peer responsible for inducting the agent:

$\mathcal{P}[\sigma]' = \mathsf{acknowledge}(\mathcal{P}[\sigma], \mu, \sigma_s)$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the inclusion of an agent* μ *into a portrayal* $\mathcal{P}$ *by a peer* σ_s*, an agent* σ *should acknowledge* μ *provided that:*

- *There exists an agent* μ *with a role in the interaction to which portrayal* $\mathcal{P}$ *is attached.*
- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *and* $\sigma_s \in \Sigma$.

If these conditions are met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma \cup \{\mu\}, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$.
2. *A message* $\mathsf{acknowledged}(\mu)$ *should be sent to agent* σ_s.

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

This ensures that agents can be added to an active portrayal process in an asynchronous distributed system without inadvertently missing any important information.

5.3.2 New Portrayable Constraints on Interaction

Whenever interaction advances, an agent can check to see if the set of portrayable propositions has changed so as to affect the argument space of the interaction portrayal. New portrayable propositions can emerge if prior constraints are resolved, new roles in interaction are adopted, new messages are received from peers or an action prescribed by the interaction protocol is executed — essentially, anything which might affect the variable space of the interaction.

Definition 5.11 *An agent* σ *with a process model* $\mathcal{M}[R, \sigma]$ *need not update the argument space* Δ *of a portrayal* $\mathcal{P}$ *as long as* $\mathcal{M}[R, \sigma]$ *does not define any new portrayable propositions:*

$$\mathsf{reportray}(\mathcal{M}[R, \sigma]) \leftarrow \left(\begin{array}{l} \mathsf{portrayal}(\sigma, \mathcal{P}[\sigma]) \wedge \mathsf{portrayables}(\mathcal{P}[\sigma], \Upsilon_{\mathcal{P}}) \wedge \\ \mathsf{portrayables}(\mathcal{M}[R, \sigma], \Upsilon_R) \wedge \Upsilon_R \subseteq \Upsilon_{\mathcal{P}} \end{array} \right)$$

Otherwise, agent σ *must extend* Δ *so that* σ *and its peers can produce arguments for possible resolutions of any new propositions:*

$$\mathsf{reportray}(\mathcal{M}[R, \sigma]) \leftarrow \left(\begin{array}{ll} \mathsf{portrayal}(\sigma, \mathcal{P}[\sigma]) \wedge \mathsf{portrayables}(\mathcal{P}[\sigma], \Upsilon_{\mathcal{P}}) & \wedge \\ \mathsf{portrayables}(\mathcal{M}[R, \sigma], \Upsilon_R) \wedge \Upsilon' = \Upsilon_{\mathcal{P}} / \Upsilon_R & \wedge \\ \mathsf{extend\_space}(\mathcal{P}[\sigma], \Upsilon') & \end{array} \right)$$

Where:

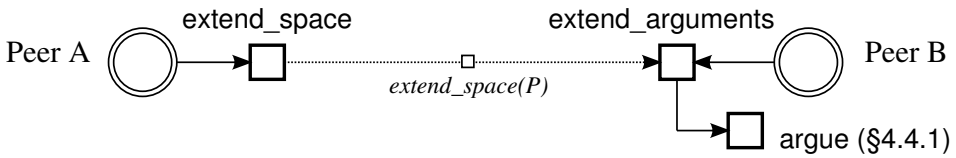


Figure 5.5: *Upon identifying additional portrayable propositions, all peers are invited to posit new arguments supporting or refuting those propositions.*

- $\mathsf{portrayal}(\sigma, \mathcal{P}[\sigma])$ *is true if* $\mathcal{P}[\sigma]$ *is the portrayal instance of* $\mathcal{P}$ *held by agent* σ.
- $\mathsf{portrayables}(\mathcal{P}[\sigma], \Upsilon)$ *is true if* Υ *is the set of portrayable propositions in* $\mathcal{P}$*, as per Definition 5.1.*
- $\mathsf{portrayables}(\mathcal{M}[R,\sigma], \Upsilon)$ *is true if* Υ *is the set of portrayable propositions in process model* $\mathcal{M}[R,\sigma]$*, as per Definition 5.6.*
- $\mathsf{extend\_space}(\mathcal{P}[\sigma], \Upsilon')$ *informs all peers* $\sigma \in \Sigma$*, where* Σ *is the set of agents with instances of* $\mathcal{P}$ *according to* $\mathcal{P}[\sigma]$ *that a set of new portrayable propositions* Υ' *has been identified, which may lead to the insertion of new arguments into* $\mathcal{P}$.

The reportray predicate invokes the extend_space operation if new portrayable propositions are encountered, which ensures that all portrayal instances are updated with respect to the extended argument space of a portrayal. This may then lead to the invocation of new arguments for or against resolutions of the new propositions identified. We can invoke reportray itself whenever we update the local interaction state as described in §2.3.2.2, modifying the step case of Definition 2.14 as shown below:

Definition 5.12 *An agent* σ *will dispatch a set of messages* $(M_o \cup M_n)$ *in response to receiving messages* M_i *as it transitions from a local interaction state* $\mathcal{S}[\sigma]_i$ *to a state* $\mathcal{S}[\sigma]_f$*:*

$$\mathsf{transition}(\mathcal{S}[\sigma]_i, M_i, (M_o \cup M_n), \mathcal{S}[\sigma]_f) \quad \leftarrow$$

$$\left(\begin{array}{ll}
\mathsf{role_model}(\mathcal{S}[\sigma]_i, \mathcal{M}[R,\sigma]_i) & \wedge \\
\mathcal{M}[R,\sigma]_i \xrightarrow{R, M_i, M_j, \mathcal{S}[\sigma], M_n} \mathcal{M}[R,\sigma]_j & \wedge \\
\mathsf{reportray}(\mathcal{M}[R,\sigma]_j) & \wedge \\
\mathsf{updated_state}(\mathcal{S}[\sigma]_i, \mathcal{M}[R,\sigma]_j, \mathcal{S}[\sigma]_j) & \wedge \\
\mathsf{transition}(\mathcal{S}[\sigma]_j, M_j, M_o, \mathcal{S}[\sigma]_f) &
\end{array}\right)$$

Where all is as described in Definition 2.14 except:

- $\mathsf{reportray}(\mathcal{M}\,[R,\sigma])$ *is true if an agent* σ *can update its portrayal instance* $\mathcal{P}\,[\sigma]$ *using role model* $\mathcal{M}\,[R,\sigma]$.

Meanwhile, the extend_space operation extends the argument space of a portrayal to include the given logical propositions, opening the portrayal to new lines of argument:

$\mathsf{extend\_space}(\mathcal{P}\,[\sigma],\Upsilon')$ — *An agent* σ *with a theory context* $\mathcal{C}$ *adds a set of logical propositions* Υ' *to the set of portrayable propositions in portrayal* $\mathcal{P}$ *if and only if:*

- Υ_R *is the set of portrayable propositions in the process model* $\mathcal{M}\,[R,\sigma]$ *of agent* σ *in a role R as per Definition 5.6.*
- $\Upsilon' = \Upsilon/\Upsilon_R$*, where* Υ *is the set of portrayable propositions in portrayal instance* $\mathcal{P}\,[\sigma]$ *and* $\Upsilon' \neq \emptyset$

A message $\mathsf{extend\_space}(\mathcal{P}\,[\sigma],\Upsilon')$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

An agent should only extend the argument space of a portrayal if recent developments in the interaction to which the portrayal is attached means that the agent is able to more precisely identify the logical constraints which must be resolved in order to complete its role in the interaction. This is determined by the reportray predicate.

Upon reception of a message $\mathsf{extend\_space}(\Upsilon')$, an agent σ with a portrayal instance $\mathcal{P}\,[\sigma]$ should invoke the extend_arguments function:

Definition 5.13 *An agent* σ *updates its portrayal instance* $\mathcal{P}\,[\sigma]$ *upon reception of a message* $\mathsf{extend\_space}(\Upsilon')$*:*

$$\mathsf{update}(\sigma,\mathcal{P}\,[\sigma],\mathcal{P}\,[\sigma]') \quad \leftarrow \quad \begin{pmatrix} \mathsf{received}(\sigma,\mathsf{extend\_space}(\Upsilon')) & \wedge \\ \mathcal{P}\,[\sigma]' = \mathsf{extend\_arguments}(\mathcal{P}\,[\sigma],\Upsilon') \end{pmatrix}$$

Where:

- $\mathsf{received}(\sigma,M)$ *is true if agent* σ *has received a message M from one of its peers in* Σ *of* $\mathcal{P}$.
- $\mathsf{extend\_arguments}(\mathcal{P}\,[\sigma],\Upsilon')$ *returns an updated portrayal instance* $\mathcal{P}\,[\sigma]$ *such that* $\Upsilon' \subseteq \Upsilon$ *of* $\mathcal{P}\,[\sigma]$.

Function extend_arguments ensures that the given propositions are added to the set of portrayable propositions recorded by an agent's portrayal instance. It also looks to see if the new portrayal space permits the addition of new arguments from an agent's theory context:

$\mathcal{P}[\sigma]' = \mathsf{extend\_arguments}(\mathcal{P}[\sigma], \Upsilon')$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the addition of new portrayable propositions* Υ' *into a portrayal* $\mathcal{P}$*, an agent* σ *should extend the portrayal space of* $\mathcal{P}$ *to include* Υ' *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *and* $(\Upsilon'/\Upsilon) \neq \emptyset$.

If this condition is met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, (\Upsilon \cup \Upsilon'), (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$.
2. *Invoke* $\mathsf{argue}(\mathcal{P}[\sigma]')$.

We define procedure argue in §5.4.1.

5.3.3 Resolving Constraints on Interaction

For any interaction augmented by an interaction portrayal, we insist that any constraints on interaction be resolved only once the portrayal becomes stable given the current set of portrayable propositions; by this, we mean that every agent in the interaction has reconciled its theory context with its portrayal instance, and thus their beliefs have been synchronised within the portrayal space:

Definition 5.14 *A portrayal* $\mathcal{P}$ *is* **stable** *if and only if for every agent* $\sigma \in \Sigma$*, where* Σ *is the set of agents possessing instances of* $\mathcal{P}$*, portrayal instance* $\mathcal{P}[\sigma]$ *is* reconciled *with* $\mathcal{C}$*, where* $\mathcal{C}$ *is the theory context of agent* σ.

Therefore there must be, in the execution model for distributed interaction, a point at which the model references the portrayal attached to interaction; this point will be immediately prior to an attempt to satisfy a logical proposition in a constraint imposed by the protocol for an interaction.

We can demonstrate when the portrayal state is referenced by the rewrite rules defined in Definition 2.15 for LCC role clauses — from this, an equivalent case can be extrapolated for any equivalent protocol language:

Definition 5.15 *A sub-clause P constrained by a logical constraint C within a role model* $\mathcal{M}[R, \sigma]$ *describing an agent* σ*'s role R in an interaction* $\mathcal{I}$ *can only be unfolded if C can be satisfied admissibly given the portrayal of* $\mathcal{I}$*:*

$$P \leftarrow C \xrightarrow{R, M_i, M_j, \mathcal{S}[\sigma], M_o} E \quad \textit{if} \quad \begin{array}{ll} \mathsf{portrayed}(\mathcal{S}[\sigma], C) & \wedge \\ \mathsf{satisfied}(\sigma, C) & \wedge \\ P \xrightarrow{R, M_i, M_j, \mathcal{S}[\sigma], M_o} E & \end{array}$$

This rewrite rule replaces the rewrite rule for constrained action in Definition 2.15. It adds a new condition which must be satisfied:

- $\mathsf{portrayed}(\mathcal{S}[\sigma], C)$ *is true if the portrayal* $\mathcal{P}$ *of the interaction described by interaction state* $\mathcal{S}$ *is* stable *as per Definition 5.14 and* C *is admissible in* $\mathcal{P}[\sigma]$.

The purpose of proposition $\mathsf{portrayed}(\mathcal{S}[\sigma], C)$ is to put interaction on hold until any arguments which might influence the resolution of constraint C have played out, temporarily synchronising the portrayal mechanism with the main interaction process.

Definition 5.16 *A logical constraint* C *has been portrayed within interaction state* $\mathcal{S}$ *if and only if* C *is admissible according to the portrayal* $\mathcal{P}$ *of the interaction described by* $\mathcal{S}$*, and* $\mathcal{P}$ *is considered stable by executing agent* σ*:*

$$\mathsf{portrayed}(\mathcal{S}[\sigma], C) \quad \leftrightarrow \quad \left(\begin{array}{l} \mathsf{portrayal}(\sigma, \mathcal{P}[\sigma]) \ \wedge \ \mathsf{stable}(\mathcal{P}[\sigma], \mathcal{P}[\sigma]') \ \wedge \\ \mathsf{admissible}(C, \mathcal{P}[\sigma]') \end{array} \right)$$

Where:

- $\mathsf{portrayal}(\sigma, \mathcal{P}[\sigma])$ *is true if* $\mathcal{P}[\sigma]$ *is the portrayal instance of* $\mathcal{P}$ *held by agent* σ.
- $\mathsf{stable}(\mathcal{P}[\sigma], \mathcal{P}[\sigma]')$ *stalls the local interaction process until agent* σ *considers* $\mathcal{P}$ *to be stable. This has no effect on other agents, nor does it prevent agent* σ *from responding to portrayal updates in the meantime or engaging in other activities not linked to this specific interaction — because the portrayal might be updated during this hiatus,* stable *returns the most current portrayal instance* $\mathcal{P}[\sigma]'$ *upon returning control.*

An agent σ can determine portrayal stability by dispatching a message stabilise to every peer $\mu \in \Sigma$ and awaiting a message stabilised from all μ. Upon receiving a message stabilise from agent σ, a peer μ is expected to respond with stabilised upon being able to confirm that its portrayal instance $\mathcal{P}[\mu]$ is reconciled with its theory context $\mathcal{C}$. Agent μ can easily determine this based on whether it is engaged in any activity involving the portrayal $\mathcal{P}$ — if μ is (in this respect) idle, then μ can respond immediately; if μ is engaged in any operations (such as reportray in §5.3 or insert in §5.4.2), or has any unresolved received messages to deal with (such as messages of type posit in §5.4.2 or observe in §5.4.6), then it must effectively clear its action queue before dispatching stabilised.

Only once stabilised has been received from all agents $\mu \in \Sigma$ can agent σ consider the portrayal to be stable.[5]

[5] Of course in an asynchronous system, it is possible that changes have occurred to the portrayal such that a peer's theory context is no longer reconciled with its portrayal instance after sending stabilised,

5.3.4 Completing Interaction

A portrayal is disposed of at the end of interaction, once every agent has completed all of its assigned roles in interaction. Until that point, all agents continue to maintain their portrayal instances, even if individually they have already completed their own contributions to the interaction being portrayed. This allows an agent to continue to provide arguments regarding constraints imposed on other agents' roles.

In order to determine when a distributed interaction has been brought to a close, each agent needs to inform its peers when it completes all of its roles in interaction; once all peers have reported this, the portrayal can be itself closed. Thus, we augment the base case of Definition 2.14:

Definition 5.17 *The local interaction state $\mathcal{S}[\sigma]$ of an agent σ cannot advance without an event to trigger that advancement. This forms a base case for local state transition:*

$$\mathsf{transition}(\mathcal{S}[\sigma], \emptyset, \emptyset, \mathcal{S}[\sigma]) \quad \leftarrow \quad \mathsf{portrayal}(\sigma, \mathcal{P}[\sigma]) \wedge \mathsf{close}(\mathcal{P}[\sigma])$$

Where:

- $\mathsf{portrayal}(\sigma, \mathcal{P}[\sigma])$ *is true if* $\mathcal{P}[\sigma]$ *is the portrayal instance of* $\mathcal{P}$ *held by agent* σ.
- $\mathsf{close}(\mathcal{P}[\sigma])$ *informs all peers* $\mu \in \Sigma$*, where* Σ *is the set of agents with instance of* $\mathcal{P}$ *according to* $\mathcal{P}[\sigma]$ *that* σ *has finished its role(s) in the interaction to which* $\mathcal{P}$ *is attached, bringing* $\mathcal{P}$ *closer to the end of its life.*

Function close simply dispatches a message closed to all peers $\mu \in \Sigma$. Once all agents $\sigma \in \Sigma$ have dispatched such a message, the interaction is determined to have ended, and the portrayal $\mathcal{P}$ is disposed of; since the information in $\mathcal{P}$ is already absorbed into the theory context of every agent in Σ, there is no further action to take. Until such an event however, all agents in Σ can contribute and draw information from $\mathcal{P}$.

It is possible, if rare, for an agent to adopt a new role in the interaction to which $\mathcal{P}$ is attached *after* dispatching a message closed, in which case the agent can cancel the closure with a message reopen dispatched to all peers.

Now that we know how to construct a portrayal and how to update it in line with the state of the interaction to which it is attached, we can now focus on how to actually generate arguments within the portrayal, and on how those arguments then influence the beliefs of agents.

---

but before resolution of constraint *C*. In an asynchronous system this scenario is unavoidable and would effectively be considered to have occurred after resolution.

5.4 Generating Arguments within a Portrayal

An agent manipulates an interaction portrayal by performing operations which dispatch messages to all peers (including itself) motivating some kind of response. These operations all have conditions applied to them that limit when they can be invoked, ensuring that only arguments and annotations of arguments justified by the portrayal space and the theory context of the given agent are inserted into the portrayal. The nature of the response to a message declaring some update of the portrayal depends on the type of message received and the state of an agent's portrayal instance at the point of reception. If a portrayal is updated with new arguments, an agent must reconcile those arguments with its own theory context. If new arguments can be drawn from that theory context in turn which now fit into the revised argument space of the portrayal, then the agent can update the portrayal accordingly, provoking more responses from peers.

Within an asynchronous system however, it cannot be ensured that messages notifying peers of updates to a portrayal will be received in the order expected. Consequently, it is necessary to ensure that operators are to all intents and purposes associative, insomuch as the order in which operations are applied is unimportant.

At the same time, all response functions are assumed to be executed in a serialised fashion such that whilst any specific function is being invoked (such as insert), any further responses to other events will be held off until the function terminates. This ensures that the integrity of the portrayal instance is maintained. In particular, a function might invoke operations such as observe and dismiss which invite response from the executing agent as well as its peers; the correct behaviour in these circumstances is to immediately return control to the function upon dispatching any required messages, and then waiting until the function itself completes before handling any self-requested response. The reader should note that all functions are written with this behaviour specifically in mind. It is also assumed that messages are responded to in the order that they are received.

5.4.1 Identifying Arguments to Insert into a Portrayal

The argue predicate tries to generate new arguments within the argument space of a portrayal using an agent's theory context. argue is most commonly invoked by reconcile in order to generate attacks against existing arguments in a portrayal (see §5.4.11 below), but is also called when the portrayal space is extended due to changes in the

state of the interaction to which a portrayal is attached (thus permitting new arguments in support of, or against, new claims; see §5.3.2) or the theory context itself changes due to outside influences:

argue($\mathcal{P}[\sigma]$) — *Given a portrayal instance $\mathcal{P}[\sigma]$, an agent σ with a theory context $\mathcal{C}$ and an unrejected extension $\mathcal{U}$ of $\mathcal{C}$ should insert new arguments into $\mathcal{P}$ if and only if:*

- *There exists an argument* $\mathbf{a} \in \Delta$*, where Δ is the argument space of $\mathcal{P}$, such that* $\mathbf{a} \notin \mathcal{P}$ *and* $\mathbf{a} \sqsubseteq \mathbf{b}$*, where* $\mathbf{b} \in \mathcal{U}$*.*

If this condition is met, then:

1. *Whilst there exists an argument* $\mathbf{c} \in \mathcal{U}$ *such that:*
   (a) $\mathbf{c} \rightharpoonup \mathbf{d}$ *for some elaboration* $\mathbf{d} \in \mathcal{C}$ *upon an argument* $\mathbf{e} \in \mathcal{P}$ *such that* $\mathbf{e} \sqsubseteq \mathbf{d}$*.*
   (b) *There does not exist an alternative elaboration* $\mathbf{f} \in \mathcal{E}$*, where $\mathcal{E}$ is the accepted extension of $\mathcal{C}$, upon* $\mathbf{e}$ *such that* $\mathbf{e} \sqsubseteq \mathbf{f} \not\sqsubseteq \mathbf{d}$ *and* $\mathbf{d} \not\sqsubseteq \mathbf{f}$*.*
   (c) *There does not exist an attack* $\mathbf{g} \rightharpoonup \mathbf{e}$ *according to $\mathcal{P}$ already, where* $\mathbf{g} \sqsubseteq \mathbf{c}$*.*

   Invoke attack$(\mathcal{P}[\sigma], \mathbf{g} \rightharpoonup \mathbf{e})$*, where* $\mathbf{g} \in \Delta$ *and* $\mathbf{g} \sqsubseteq \mathbf{c}$*.*
2. *Whilst there exists an argument* $\mathbf{c} \in \mathcal{U}$ *such that:*
   (a) *There does not exist an argument* $\mathbf{d} \in \mathcal{P}$ *such that* $\mathbf{d} \sqsubseteq \mathbf{c}$*.*
   (b) *There does exist an argument* $\mathbf{e} \in \Delta$*, where Δ is the argument space of $\mathcal{P}$, such that* $\mathbf{e} \sqsubseteq \mathbf{c}$*.*

   Invoke posit$(\mathcal{P}[\sigma], \mathbf{e})$*.*

If this condition is not *met, then do nothing.*

Function argue invokes a number of instances of the posit (§5.4.2) and attack (§5.4.4) operations. Operation posit asserts new arguments into a portrayal where deemed to be within the portrayal space as per Definition 5.2. Operation attack asserts attacks relations between arguments within a portrayal. Notably, attack can subsume the initial positing of the attack argument and can also elaborate upon a target argument already within the portrayal so as to ensure that it is clearly attacked by the chosen attacking argument. Thus it is only necessary to directly invoke posit if directly claiming a resolution of a portrayable proposition (rather than trying to undercut or rebut an existing

argument), or if producing an alternative elaboration of a potential argument within the portrayal which has already been elaborated upon as a distinct argument (in order to restore the admissibility of a claim in the presence of attacks against other supporting arguments; see §5.1.3 and §5.4.2 below).

'Outside influences' include any event not part of the portrayal mechanism; such events include actions taken as part of the interaction to which interaction is attached, events which are part of other interactions and changes in agent or environment state. In essence, anything that might change the context of the theory with which an agent may make decisions in interaction, and thus which might affect the generation and interpretation of arguments within a portrayal, can cause an invocation of the argue function. Given that this regards an agent's personal theory context, and is not caused by the portrayal mechanism, it is up to the individual autonomous agent to monitor such events and invoke argue at its own discretion.

The complexity of generating arguments using argue is tied to the underlying mechanics of the theory context from which arguments are drawn; in essence, an agent is simply looking for arguments already privately evaluated which will affect the state of the portrayal given its current argument space. For the most part:

- If argue is invoked by reconcile, then it will produce an attack against the most recent argument inserted into the portrayal, or, if an argument has been elaborated upon in order to be attacked by another agent, it will re-establish support for that argument's claim by positing an alternative argument with that same claim.

- If argue is invoked by extend_arguments (§5.3.2), then it will posit a few basic arguments claiming specific instantiations of any new portrayable propositions.

- If argue is invoked by a change in the environment, then it will generally only produce a single argument, either as an attack against a prior argument, or as a new claim regarding the resolution of a constraint on interaction.

Thus, whilst the argue procedure can in theory simultaneously generate an array of arguments making new claims, attacking existing arguments and positing alternative elaborations upon existing arguments, in practice argue usually only inserts one or two arguments at a time, and will often add no new arguments at all. The most common exception is in the case where an agent is newly added to an already quite advanced interaction, in which case the agent may be able to make several new arguments in response to those already present.

Example 5.6 *Continuing on from Example 5.5, let us assume that Alanna determines that* $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ *and* $\mathsf{patron}(\mathsf{benjamin},\mathsf{library})$. *The immediate consequence of this is that Benjamin in the role of* advocate *is sent a* request *message by Alanna. In doing this, Alanna is also compelled to send on to Benjamin a copy of the portrayal instance* $\mathcal{P}[\mathsf{alanna}]$ *as specified in §5.3.1.*

Benjamin then has to reconcile the content of $\mathcal{P}[\mathsf{benjamin}]$ *(his copy of* $\mathcal{P}[\mathsf{alanna}]$*) as specified by procedure* validate. *Assume that after executing* reconcile *(see §5.4.11), Benjamin is able to produce the following potential arguments:*

$$\begin{aligned}\mathbf{a}'' = \langle\ \{\ & \mathsf{source}(\mathsf{library},\mathsf{archives}),\\ & \neg\mathsf{read\_data}(\mathsf{alanna},\mathsf{archives}),\\ & \forall X,Y,Z.\ \mathsf{accessible}(X,Y)\wedge\mathsf{source}(Y,Z)\rightarrow\mathsf{read\_data}(X,Z)\ \},\\ & \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\\ \mathbf{e}' = \langle\ \{\ & \mathsf{read\_data}(\mathsf{alanna},\mathsf{archives})\ \},\ \mathsf{read\_data}(\mathsf{alanna},\mathsf{archives})\ \rangle\end{aligned}$$

In this case, Benjamin can attack argument $\mathbf{a}'$ *if he assumes that Alanna's reasoning follows* $\mathbf{a}''$ *and then argues* $\mathbf{e}'$. *Given that Benjamin does not (as yet) know of an alternative means of deriving* $\mathbf{a}'$*, invoking* argue *will in turn invoke* $\mathsf{attack}(\mathcal{P}[\mathsf{benjamin}],\mathbf{e}' \rightharpoonup \mathbf{a}'')$.

Leaving aside Alanna's response for now, the adoption of role advocate *by Benjamin produces new portrayable propositions;* controller(*Controller*, library) *and* trustworthy(alanna, access(library)). *Thus we can expect the* extend_space *operation to be invoked by Benjamin (§5.3.2), which will lead to* argue *being invoked by both Alanna and Benjamin. If we assume that Alanna can produce the following potential argument:*

$$\mathbf{f}' = \langle\ \{\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$$

...and if we assume that Benjamin can produce the following argument:

$$\mathbf{g}' = \langle\ \{\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \},\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \rangle$$

...then $\mathsf{posit}(\mathcal{P}[\mathsf{alanna}],\mathbf{f}')$ *and* $\mathsf{posit}(\mathcal{P}[\mathsf{benjamin}],\mathbf{g}')$ *will be invoked by Alanna and Benjamin respectively.*

Finally, let us skip ahead. Assume that Charlotte has accepted Benjamin's recommendation, and has granted Alanna access to library. *By protocol* acquire_access, *Charlotte enacts* $\mathsf{permit}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))$, *which changes the environment into one in which Alanna has the desired access. This invokes* argue *yet again, this time allowing Alanna to defeat her own claim* $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ *and thus demonstrate to herself the success of the interaction she started.*

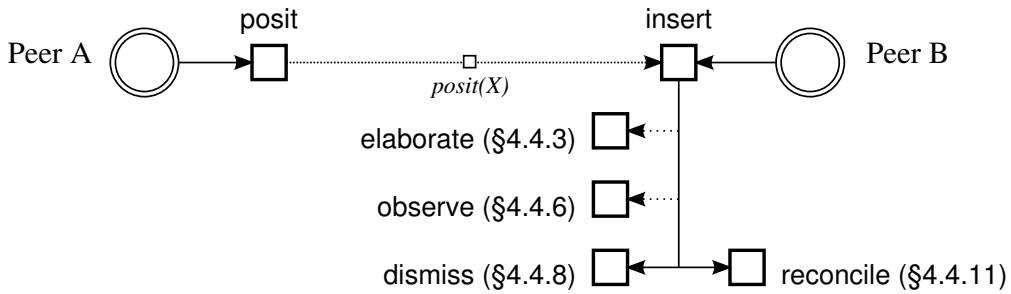


Figure 5.6: *Upon the positing of an argument by Peer A, each agent should make any relevant observations before either dismissing the argument as invalid or reconciling it with its own beliefs.*

5.4.2 Positing Arguments into a Portrayal

The posit operation is used to add new arguments to a portrayal.

$\mathsf{posit}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle)$ — *An agent* σ *with a theory context* $\mathcal{C}$ *and an unrejected extension* $\mathcal{U}$ *of* $\mathcal{C}$ *posits a new argument* $\langle \Phi, \alpha \rangle$ *into portrayal* $\mathcal{P}$ *if and only if:*

- $\Phi \vdash \alpha$ *and* $\langle \Phi, \alpha \rangle$ *is minimal (i.e. there exists no subset* $S \subset \Phi$ *such that* $S \vdash \alpha$*) according to* $\mathcal{C}$*.*
- *There exists an argument* $\mathbf{a} \in \mathcal{U}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{a}$*.*
- *There does not exist an argument* $\mathbf{b} \in \mathcal{P}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{b}$ *or* $\mathbf{b} \sqsubseteq \langle \Phi, \alpha \rangle$*.*
- $\langle \Phi, \alpha \rangle \in \Delta$*, where* Δ *is the argument space of* $\mathcal{P}$*.*

A message $\mathsf{posit}(\langle \Phi, \alpha \rangle)$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$ *(including agent* σ*).*

The posit operation ensures that agents only posit arguments which are relevant to the interaction being enacted and which they have not already privately rejected.

An agent should refrain from knowingly positing into a portrayal $\mathcal{P}$ an argument **a** which it believes to potentially be an existing argument **b** already present in $\mathcal{P}$: if an agent can formulate another argument **c** such that $\mathbf{a} \sqsubset \mathbf{c}$ which is distinct from **b** (such that $\mathbf{c} \not\sqsubseteq \mathbf{b}$ and $\mathbf{b} \not\sqsubseteq \mathbf{c}$, and thus **c** and **b** are separate arguments which happen to share **a** as a potential argument), then it should posit **c** — otherwise it has no basis on which to justify positing a new argument.

Likewise, if an agent wishes to posit an argument **a** for which an existing potential argument **b** exists, then that agent should perform an *elaborate* operation on **b** rather

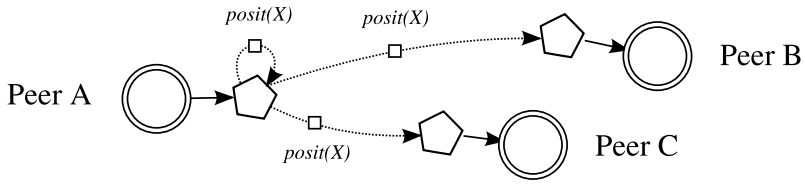
than positing a new argument; it is worth noting that if **b** has alternative elaborations known to other peers, then those peers can always posit any alternative elaboration as a distinct argument as described just above.

Finally, a posited argument should be within the argument space of the portrayal. Note that whilst an argument **a** can be within the argument space Δ of a portrayal $\mathcal{P}$ if it attacks another argument **b** already in $\mathcal{P}$, it is necessary to perform an attack operation in order to formally assert the attack relation $\mathbf{a} \rightharpoonup \mathbf{b}$ in $\mathcal{P}$ (regardless, **a** would still be within Δ).

Provided that all conditions are fulfilled, an agent will send a message $\mathsf{posit}(\mathbf{a})$ to all agents possessing instances of portrayal $\mathcal{P}$. Given the positing of a new argument by an agent, there are three basic responses which its peers can immediately make:

Insert a New Argument — If the posited argument is genuinely new, such that it adds to the information expressed by the portrayal, then the recipient should add it to its portrayal instance:

Peer A posits a new argument X to peers B, C and itself:

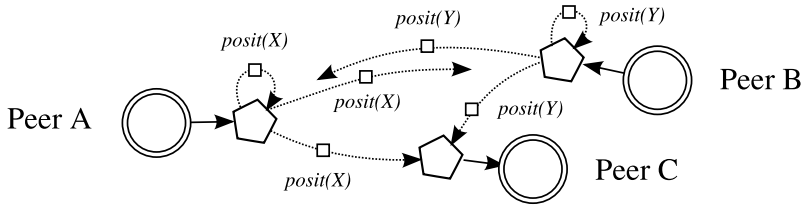


If argument X is not present at all in an agent's portrayal instance, then it will be inserted into it.

The agent which posited the argument will naturally respond to its own action in this fashion.

Expand Existing Arguments — Any delay propagating existing arguments to all agents may lead to an agent positing what it (wrongly) thinks to be an original argument. If the posited argument is merely an elaboration upon existing arguments in the portrayal, then the recipient should simply replace those existing arguments with the posited one:

Peer A posits an argument X, unaware that peer B has also posited an argument Y:



> *If $Y \sqsubset X$, then peer C, receiving Y first, will insert Y in its portrayal instance and then expand it into X.*

It is possible (if rare) for more than one argument in a portrayal to potentially be the same elaboration, in which case all such potential arguments are simply replaced by the single elaboration.

Ignore the Posited Argument — If the posited argument is already present in the portrayal, or if it is potentially an existing argument, then the posited argument is redundant and can be ignored:

> *Returning briefly to the example of the previous case, if $X \sqsubseteq Y$, then peer C will insert argument Y into its portrayal instance and then ignore X afterwards.*

This response might occur if, for example, two separate agents both posit the same argument simultaneously.

In order to solicit the correct one of these responses, upon reception of a message $\mathsf{posit}(\mathbf{a})$, an agent σ with a portrayal instance $\mathcal{P}[\sigma]$ should invoke the insert function:

Definition 5.18 *An agent σ updates its portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message $\mathsf{posit}(\mathbf{a})$:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{posit}(\mathbf{a})) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a}) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M)$ *is true if agent σ has received a message M from one of its peers in Σ of $\mathcal{P}$.*
- $\mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a})$ *returns an updated portrayal instance $\mathcal{P}[\sigma]$ such that $\mathbf{a} \in \mathcal{P}$.*

Function insert serves to ensure that a given argument is represented within the portrayal, and checks whether or not it is valid, as well as whether there are any additional observations to be made:

$\mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle)$ — *Given an existing portrayal instance $\mathcal{P}[\sigma]$ and in response to an argument $\langle \Phi, \alpha \rangle$, a peer σ with theory context $\mathcal{C}$ should insert $\langle \Phi, \alpha \rangle$ into $\mathcal{P}$ provided that:*

- *There exists no argument $\mathbf{a} \in \mathcal{P}$ such that $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{a}$.*

- *S is the set of all arguments* $\mathbf{b} \in \mathcal{P}$ *such that* $\mathbf{b} \sqsubset \langle \Phi, \alpha \rangle$.
- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$.

If these conditions are met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}', \rightharpoonup), \mathsf{inv}'_{\mathcal{A}}, \mathsf{inv}'_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$, *where:*
   (a) $\mathcal{A}' = (\mathcal{A}/S) \cup \{\langle \Phi, \alpha \rangle\}$.
   (b) *For every agent* $\mu \in \Sigma$, *if* $\mathbf{a} \in \mathsf{inv}_{\mathcal{A}}(\mu)$ *for some argument* $\mathbf{a} \in S$, *then* $\mathbf{a} \notin \mathsf{inv}'_{\mathcal{A}}(\mu)$; *otherwise if* $\mathbf{a} \in \mathsf{inv}_{\mathcal{A}}(\mu)$, *then* $\mathbf{a} \in \mathsf{inv}'_{\mathcal{A}}(\mu)$.
   (c) *If* $\mathbf{a} \rightharpoonup \mathbf{b}$, *where* $\mathbf{a} \in S$ *and* $\mathbf{b} \in \mathcal{A}$, *then* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{b}$; *if* $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\mu)$ *for any peer* $\mu \in \Sigma$, *then* $(\langle \Phi, \alpha \rangle, \mathbf{b} \in \mathsf{inv}'_{\rightharpoonup}(\mu)$.
   (d) *If* $\mathbf{b} \rightharpoonup \mathbf{a}$, *where* $\mathbf{a} \in S$ *and* $\mathbf{b} \in \mathcal{A}$, *then* $\mathbf{b} \rightharpoonup \langle \Phi, \alpha \rangle$; *if* $(\mathbf{b}, \mathbf{a} \in \mathsf{inv}_{\rightharpoonup}(\mu)$ *for any peer* $\mu \in \Sigma$, *then* $(\mathbf{b}, \langle \Phi, \alpha \rangle) \in \mathsf{inv}'_{\rightharpoonup}(\mu)$.
   (e) *Otherwise, if* $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\mu)$, *then* $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}'_{\rightharpoonup}(\mu)$ *for all peers* $\mu \in \Sigma$.
2. *If an elaboration* $\langle \Psi, \alpha \rangle \in \mathcal{C}$ *of* $\langle \Phi, \alpha \rangle$ *is dismissed as per Definition 3.15 because of a sentence* φ *such that* $\Theta \vdash \varphi$ *and* $\Psi \vdash \neg\varphi$, *and there exists no alternative elaboration* $\mathbf{a} \in \mathcal{U}$, *where* $\mathcal{U}$ *is the unrejected extension of* $\mathcal{C}$, *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{a}$:
   - *There exists an argument* $\langle \Psi', \alpha \rangle$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi', \alpha \rangle \sqsubseteq \langle \Psi, \alpha \rangle$ *and* $\Psi' \vdash \neg\varphi$, *but there is no argument* $\langle \Psi'', \alpha \rangle$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi'', \alpha \rangle \sqsubset \langle \Psi', \alpha \rangle$ *and* $\Psi'' \vdash \neg\varphi$.
   - *If* $\langle \Psi', \alpha \rangle = \langle \Phi, \alpha \rangle$, *then invoke* $\mathsf{observe}(\mathcal{P}[\sigma]', \{\varphi\})$.
   - *Otherwise, invoke* $\mathsf{elaborate}(\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle, \langle \Psi', \alpha \rangle)$ *before then invoking* $\mathsf{observe}(\mathcal{P}[\sigma]', \{\varphi\})$.
3. *If* $\langle \Phi, \alpha \rangle$ *is invalid with respect to* $\mathcal{C}$ *as per Definition 4.11, then invoke* $\mathsf{dismiss}(\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle)$; *otherwise invoke* $\mathsf{reconcile}(\mathcal{P}[\sigma]', \{\langle \Phi, \alpha \rangle\})$.

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

Any attacks involving potential arguments are inherited by their elaborations, as are any dismissals of potential attacks specified by the invalid argumentation function $\mathsf{inv}_{\rightharpoonup}$. Any dismissals of potential arguments are discarded on the assumption that if $\langle \Phi, \alpha \rangle$ is still considered to be invalid by a peer $\mu \in \Sigma$, then σ will receive a new message $\mathsf{dismiss}(\langle \Phi, \alpha \rangle)$ (see §5.4.8). Agent σ will also invoke the dismiss if it believes

$\langle\Phi,\alpha\rangle$ is invalid and observe (§5.4.6) if it believes the argument can be dismissed by observation (σ will also minimally elaborate upon $\langle\Phi,\alpha\rangle$ if necessary to illustrate how the observation conflicts with the argument). Note that if $\langle\Phi,\alpha\rangle$ is indeed invalid according to σ, then this fact is *not* added to $\mathsf{inv}_{\mathcal{A}}(\sigma)$ at this point — instead, operation dismiss is invoked, and $\mathsf{inv}_{\mathcal{A}}(\sigma)$ will be updated in response to that. Similarly, any observations or elaborations occur in response to their specific update operations. If $\langle\Phi,\alpha\rangle$ *is* valid, then σ will invoke reconcile, which will integrate $\langle\Phi,\alpha\rangle$ into σ's theory context and make counter-arguments as necessary (§5.4.11).

5.4.3 Elaborating Upon Arguments in a Portrayal

The elaborate operation is used to elaborate upon arguments in a portrayal where existing arguments are no longer considered to be sufficiently detailed to describe the conflicts affecting an interaction.

elaborate$(\mathcal{P}[\sigma],\mathbf{a},\langle\Phi,\alpha\rangle)$ — *An agent σ with a theory context $\mathcal{C}$ and an unrejected extension $\mathcal{U}$ of $\mathcal{C}$ elaborates upon an argument* **a** *within portrayal $\mathcal{P}$, producing an expanded argument $\langle\Phi,\alpha\rangle$, if and only if:*

- $\mathbf{a}\in\mathcal{P}$ *and* $\mathbf{a}\sqsubset\langle\Phi,\alpha\rangle$.
- $\Phi\vdash\alpha$ *and* $\langle\Phi,\alpha\rangle$ *is minimal (i.e. there exists no subset* $S\subset\Phi$ *such that* $S\vdash\alpha$*) according to* $\mathcal{C}$.
- *There is an admissible argument* $\mathbf{b}\in\mathcal{U}$ *such that* $\langle\Phi,\alpha\rangle\sqsubseteq\mathbf{b}$.
- *Either:*
  - $\langle\Phi,\alpha\rangle\in\Delta$*, where* Δ *is the argument space of* $\mathcal{P}$.
  - $\Phi\vdash\varphi$ *and* $\Theta\vdash\neg\varphi$*, where* Θ *is the theory core of* $\mathcal{C}$*, and there exists no argument* $\langle\Phi',\alpha\rangle$ *such that* $\Phi'\vdash\varphi$ *and* $\langle\Phi',\alpha\rangle\sqsubset\langle\Phi,\alpha\rangle$ *(see §5.4.6).*
  - elaborate *has been invoked in response to an* inquire *operation (see §5.4.5).*

A message elaborate$(\mathbf{a},\langle\Phi,\alpha\rangle)$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

An agent can elaborate upon any argument in a portrayal for which it knows of a more concrete formulation if it has a justification for inserting that more concrete formulation into the portrayal. An agent is justified in elaborating upon an argument if either:

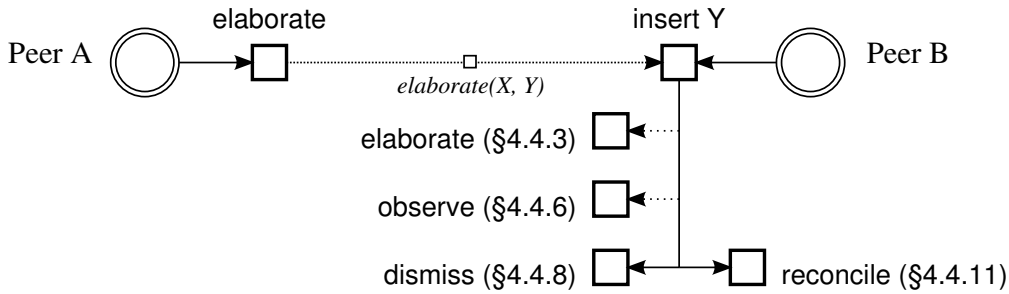


Figure 5.7: *Upon the elaboration of an argument by Peer A, each agent should make any relevant observations before either dismissing the elaboration as invalid or reconciling it with its own beliefs.*

- There exists an argument $\mathbf{a} \in \mathcal{P}$ such that $\mathbf{a} \notin \Delta$ with respect to the theory context $\mathcal{C}$ of an agent σ; this can happen if another argument **b** is added to $\mathcal{P}$ such that **b** attacks all elaborations of **a** in $\mathcal{E}$ of $\mathcal{C}$, where $\mathcal{E}$ is the accepted extension of $\mathcal{C}$ (i.e. if σ believes that **a** can only be justified if **b** is defeated within the portrayal, but **a** is not expressed to a sufficient level of detail to show that $\mathbf{b} \rightharpoonup \mathbf{a}$), in which case σ can elaborate upon **a** to rectify this situation.

- There exists an argument $\mathbf{a} \in \mathcal{P}$ such that for all elaborations **b** of **a** in an agent σ's theory context $\mathcal{C}$, it is the case that $\mathbf{c} \rightharpoonup \mathbf{b}$ for some argument $\mathbf{c} \in \mathcal{U}$, where $\mathcal{U}$ is the accepted extension of $\mathcal{C}$, in which case σ will want to attack **a** using some potential argument of **c**. To do this however in a manner considered valid by its peers, agent σ may have to elaborate upon **a** such that it exhibits a vulnerability to **c** (i.e. the elaboration of **a** must be supported by a proposition directly contradicted by the claim of **c** as per the definition of attack in Definitions 3.10).

- There exists an observation $\varphi \in \Theta$, where Θ is the theory core of theory context $\mathcal{C}$, such that all elaborations **b** of **a** in $\mathcal{E}$ (the accepted extension of $\mathcal{C}$) infer $\neg\varphi$, in which case an agent believes that **a** should be dismissed; first however, **a** must be elaborated upon to the point that it itself infers $\neg\varphi$.

- Agent σ has received a message $\mathsf{inquire}(\mathbf{a})$ such that a peer has declared an interest in the reasoning underlying a, and σ is able to comply (§5.4.5).

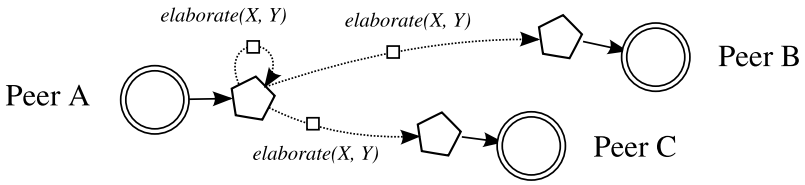
Oddly enough, the first two reasons to elaborate upon arguments are generally *not* handled directly by the elaborate operation. Instead, an agent can invoke attack, which will elaborate upon the defending argument directly so as to illustrate the attack relation being posited (see §5.4.4). Given that invoking attack is more efficient than

invoking elaborate and *then* attack, and that the argue predicate described in §5.4.1 will invoke attack in such circumstances, the elaborate operation itself can be expected to only be used for the latter two cases. Even then, the third case is activated in parallel with an observe operation, and the final case is activated in response to an inquire operation.

Nevertheless, in the case of the third example (inquiry), an agent will send a message elaborate$(\mathbf{a}, \mathbf{b})$ to all agents possessing instances of a portrayal $\mathcal{P}$. The responses an agent can make to the elaboration of existing arguments in a portrayal are in truth almost identical to the responses an agent can make to the positing of new arguments. This is due to the nature of communication between three or more agents in an asynchronous distributed system, where messages can arrive out of their expected order due to arbitrary delays afflicting the medium through which dialogue is conducted. All that really changes are the particular scenarios in which certain responses might occur:

Expand Existing Arguments — If the elaboration received is a valid elaboration of existing arguments in the portrayal , then the recipient should simply replace those existing arguments with their elaboration:

Peer A elaborates upon an argument X to peers B, C and itself:

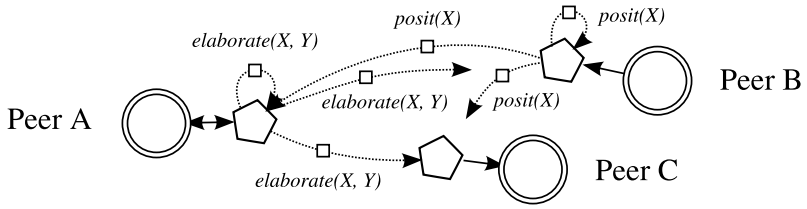


Provided that argument X is present in an agent's portrayal instance, then it will be replaced by argument Y.

As with the equivalent response to a *posit* operation (in §5.4.2), it is possible for many arguments to expand into a single concrete argument — not just the argument identified in the original message. In fact, it may even be that the intended potential argument is not available to be replaced, due perhaps to delays in the propagation of arguments between portrayal instances. The agent which decides to elaborate an argument will naturally respond to its own own action by expanding that argument within its own portrayal instance.

Insert a New Argument — If the argument to be elaborated is not present in a portrayal, and moreover if there exist no alternative arguments to elaborate in its stead, then the recipient should treat the elaboration received as if it was a new argument:

Peer A elaborates upon an argument posited by peer B:

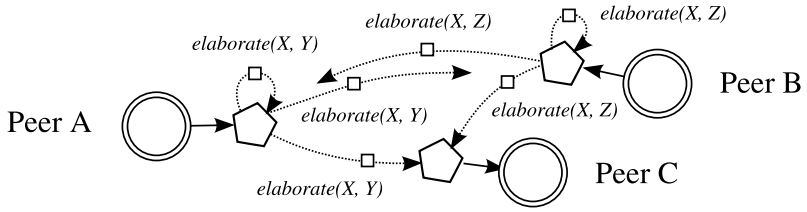


If elaboration Y reaches peer C before argument X, then C will be insert Y into its portrayal instance and then ignore X.

This response might occur if, for example, an agent elaborates on an argument posited by another, whereupon a third agent, by the vagaries of asynchronous networking, receives notification of the elaboration before notification of original posit. When the original posit is received, it will be ignored, being now merely a potential argument for an existing argument in the portrayal.

Ignore the Elaboration — Finally, if the elaboration received is already present in the portrayal, or if it is merely a potential argument for another argument already present in the portrayal, then the elaboration is redundant and can be ignored, just as for the equivalent case in §5.4.2:

Peer A elaborates upon argument X, as does peer B:



Assuming that peer C receives A's elaboration Y first, then C will either replace it with B's elaboration Z if $Y \sqsubset Z$, or ignore Z if $Z \sqsubseteq Y$.

As illustrated above, typically this occurs because two separate agents have both elaborated upon the same argument simultaneously.

Upon reception of a message $\mathsf{elaborate}(\mathbf{a}, \mathbf{b})$, an agent σ with a portrayal instance $\mathcal{P}[\sigma]$ should invoke the insert function just as in §5.4.2:

Definition 5.19 *An agent σ updates its portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message* $\mathsf{elaborate}(\mathbf{a})$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{elaborate}(\mathbf{a}, \mathbf{b})) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \mathbf{b}) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M)$ *is true if agent* σ *has received a message* M *from one of its peers in* Σ *of* $\mathcal{P}$.

- $\mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a})$ *returns an updated portrayal instance* $\mathcal{P}[\sigma]$ *such that* $\mathbf{a} \in \mathcal{P}$.

Note that despite elaborating upon a specific argument **a**, we do not appear to refer to it within function insert. This is an artefact of the specification of insert, wherein we recognise the possibility that multiple arguments might be elaborated into the same elaboration, or that there will exist an argument to be elaborated upon in a portrayal instance which is not in fact the argument expected. In practice however, one can expect that **a** is within the set S of arguments to be replaced in insert (if not S in its entirety), and that actually informing one's peers of the specific argument intended to be replaced might save (a small amount of) computation.

5.4.4 Attacking Arguments Within a Portrayal

The attack operation is used to identify (and introduce) conflicts within a portrayal, being generally invoked by the argue function (§5.4.1):

$\mathsf{attack}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle)$ — *An agent* σ *with a theory context* $\mathcal{C}$ *and an unrejected extension* $\mathcal{U}$ *of* $\mathcal{C}$ *attacks an argument* $\langle \Psi, \beta \rangle$ *within portrayal* $\mathcal{P}$ *if and only if:*

- *There exist arguments* $\mathbf{a}, \mathbf{b} \in \mathcal{C}$ *such that* $\mathbf{a} \in \mathcal{U}$*, both* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{a}$ *and* $\langle \Psi, \beta \rangle \sqsubseteq \mathbf{b}$*, and it is the case that* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *because* $\Psi \vdash \gamma$ *and* $\{\alpha\} \vdash \neg\gamma$ *for some sentence* $\gamma \in \mathcal{L}$ *according to* $\mathcal{C}$.
- *There exists an argument* $\mathbf{c}$ *such that* $\mathbf{c} \sqsubseteq \mathbf{d}$ *for some argument* $\mathbf{d} \in \mathcal{P}$ *and:*
  - $\mathbf{c} \sqsubseteq \langle \Psi, \beta \rangle$.
  - *There does not exist an argument* $\mathbf{e} \in \mathcal{E}$*, where* $\mathcal{E}$ *is the accepted extension of* $\mathcal{C}$*, such that* $\mathbf{c} \sqsubseteq \mathbf{e}$ *and* $\langle \Phi, \alpha \rangle \not\rightharpoonup \mathbf{e}$.
  - *There also does not exist an argument* $\mathbf{f}$ *such that* $\mathbf{c} \sqsubseteq \mathbf{f} \sqsubset \langle \Psi, \beta \rangle$ *and* $\langle \Phi, \alpha \rangle \rightharpoonup \mathbf{f}$.
- *There is no existing argument* $\mathbf{g} \in \mathcal{P}$ *such that* $\langle \Phi, \alpha \rangle \sqsubset \mathbf{g}$ *or* $\mathbf{g} \sqsubset \langle \Phi, \alpha \rangle$.
- $\langle \Phi, \alpha \rangle \in \Delta$*, where* Δ *is the argument space of* $\mathcal{P}$ *including argument* $\langle \Psi, \beta \rangle$.
- *There do not exist any arguments* $\mathbf{h}, \mathbf{i} \in \mathcal{P}$ *such that* $\mathbf{i} \rightharpoonup \mathbf{h}$ *according to* $\mathcal{P}$ *and* $\langle \Psi, \beta \rangle \sqsubseteq \mathbf{h}$*,* $\mathbf{h} \sqsubseteq \langle \Psi, \beta \rangle$*,* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{i}$ *or* $\mathbf{i} \sqsubseteq \langle \Phi, \alpha \rangle$.

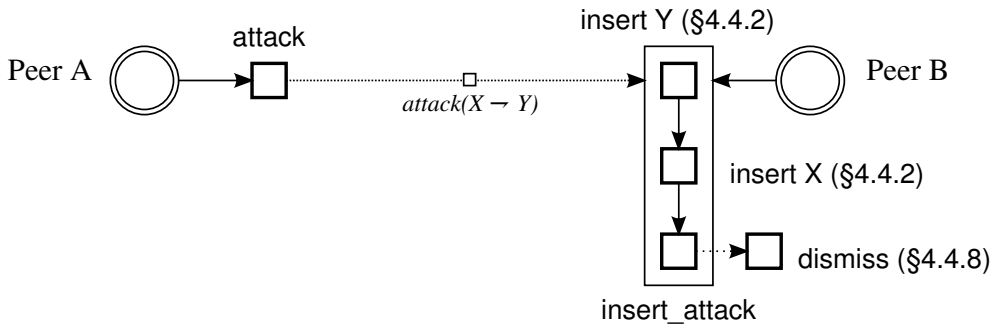


Figure 5.8: *Upon Peer A attacking an argument within the portrayal, each agent should ensure that both target and attacking arguments are inserted into their portrayal instances, and that the attack itself is recorded; if the attack is considered to be invalid, agents can dismiss it.*

A message $\mathtt{attack}(\langle\Phi,\alpha\rangle \rightharpoonup \langle\Psi,\beta\rangle)$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

Attacks are motivated by an agent's awareness of counter-arguments (described within its theory context) which have not yet been applied to arguments in the portrayal.

It is necessary for there to be an argument in the portrayal to attack. It may be necessary though for the argument to be elaborated upon before it can be undercut by the intended attack — in such a case, attack automatically performs the elaboration as part of the attack operation. Any elaboration should be the minimum required for a valid attack. However, should an agent be aware that there exist more than one distinct elaborations of an argument (i.e. at least two elaborations where no elaboration is potentially another), and at least one of those elaborations is accepted by the agent, then that agent should 'give the benefit of the doubt' and refrain from attacking (because if any of the elaborations of a potential argument is admissible, then the potential argument is admissible). Given a choice of elaborations, all of which an agent can defeat but all of which involve different attacks, the agent should simply select one arbitrarily and attack it. If another peer still accepts a different elaboration of an attacked argument to the one selected, then it is now free to posit the alternative elaboration (as noted in §5.1.3 and §5.4.2). Of course, that alternative elaboration may simply get attacked again.

One hypothetical circumstance exists where an agent might be able to attack an argument which shares a common potential argument with an argument already in the portrayal; this covers the scenario in which an argument has been elaborated in order to be attacked, but all attacks are later defeated, and yet an alternative elaboration can be

attacked as well. Note however that if the original elaboration remains undefeated, then its claim will still follow, regardless of the inadmissibility of alternative elaborations, and thus this circumstance does *not* permit the addition of alternative elaborations to the portrayal, given that it has no bearing on the ultimate admissibility of the claims for which the portrayal exists to test.

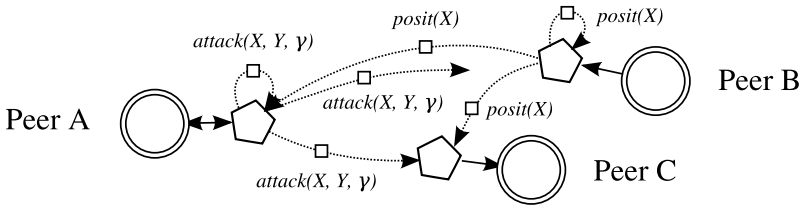
It is *not* strictly necessary for the attacking argument to be already present in the portrayal, though it can be. In practice, it is expected that most new arguments will be added to the portrayal as a side-effect of an attack operation, as opposed to a posit, which will mainly be used to introduce new resolutions for interaction constraints. The incidental positing of new arguments is still subject to the requirements of posit (§5.4.2) however — these requirements are duly subsumed by the above specification.

Finally, any new attack must not already be known to exist in the portrayal; this includes variants of the attack for different potential restrictions of the conflicting arguments. If an agent wishes for some reason to change the structure of the arguments involved in an existing attack, then the agent should elaborate upon those arguments subject to the constraints of the elaborate operation. Attack relations between arguments are passed on to elaborations of those arguments, as is evidenced by Proof 4.1.

Provided then that all conditions necessary to launch an attack are fulfilled, an agent will send a message $\mathsf{attack}(\langle\Phi,\alpha\rangle \rightharpoonup \langle\Psi,\beta\rangle)$ to all peers possessing instances of portrayal $\mathcal{P}$. The responses available to a peer depend on the state of both the attacking and defending argument within its portrayal instance:

Do Nothing with the Defending Argument — If the argument being attacked is already present within the portrayal, or if it is merely a potential argument for another argument already present in the portrayal, then there is no need to add it to the portrayal:

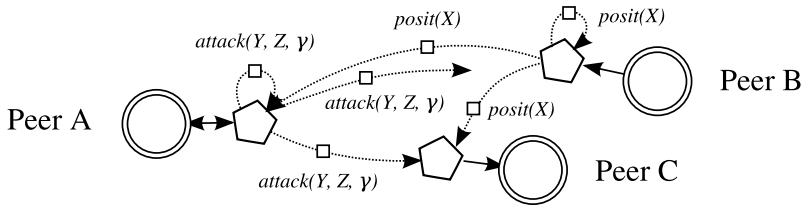
Peer A attacks an argument X posited by peer B:



If peer C receives the attack after receiving argument X from B, then peer C will have already (one way or another) inserted X into its portrayal instance.

This is the default case.

Expand Existing Arguments for Attack — If the argument being attacked is a valid elaboration of existing arguments in the portrayal, then the recipient should simply replace those existing arguments with the defending argument:

> *Rather than attacking a posited argument X, peer A attacks an elaboration Y of X:*
>
> 
>
>
> *Assuming peer C has received the original argument X prior to receiving the attack against Y, then C will simply expand X into Y prior to considering the attack proper, as will peer B later.*

As discussed in §5.4.3 and this section, if there exists an argument in the portrayal which lacks the concrete detail to be undercut by an attack, then the attacking agent can resolve this by elaborating upon it specifically to attack it; remember that peers can always invoke a posit operation to add an alternative elaboration as a distinct argument if one should be available to restore the admissibility of the attacked claim (§5.4.2). This case may also occur if, for at least one peer, the attack against a prior elaboration of an argument arrives before notification of the elaboration itself.

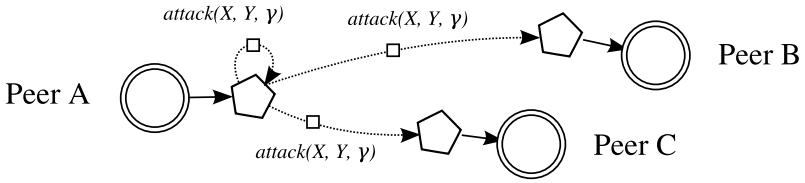
Insert a new Argument to Attack — If the defending argument is not present in the portrayal, and moreover if there exist no alternative arguments to elaborate in its stead, then the recipient should simply add the defending argument:

> *Returning to the previous example, if peer C had* not *received argument X prior to receiving the attack against its elaboration Y, then C will insert Y into its portrayal instance and later ignore the positing of X.*

This is another response which may occur if delays in communication in an asynchronous system lead to messages being received by a peer out of order.

Insert the Attacking Argument — If the attacking argument is not yet present in the portrayal, and moreover if there exist no alternative arguments to elaborate in its stead, then the recipient should add it to the portrayal:

> *Peer A can attack an argument X with a new argument Y should Y be in the argument space and capable of defeating X:*

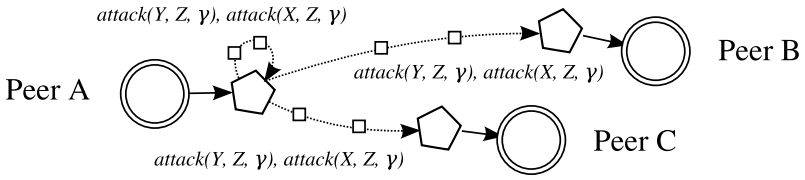


If argument Y is not present at all in an agent's portrayal instance, then it will be inserted into it.

This is how many arguments are posited in practice — as attacks against existing arguments. Therefore this is considered to constitute the default case.

Do Nothing with the Attacking Argument — If the attacking argument is present within the portrayal, or if it is merely a potential argument for another argument already present in the portrayal, then there is no need to add it to the portrayal:

Peer A can use an argument Z to attack X and then attack Y with the same argument:

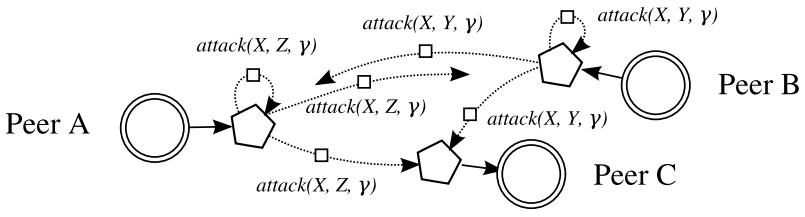


Since argument Z will be added to every agent's portrayal instance due to the first attack (provided the attacks arrive in order), it will not need to be inserted again.

This response may occur if another agent has independently made the same attack as the original attacker, if the attacking argument has been previously posited without the realisation that it constitutes an attack on another argument, or if it has been posited as part of an attack against another argument entirely.

Expand Existing Arguments to Attack With — If the attacking argument is a valid elaboration of existing arguments in the portrayal, then the recipient should simply replace those existing arguments with the attacking argument:

Peer A attacks an argument X using argument Z, unaware that peer B has already attacked X, albeit with an argument Y such that $Y \sqsubset Z$:



If peer C receives A's attack after that of B, then C will expand Y into Z in order to keep all portrayal instances synchronised. Peer B will do likewise.

This is another response which may occur if delays in communication in an asynchronous system lead to messages being received by a peer out of order (since conflicts are focused on the claim of the attacking argument against some element of the attacked argument, elaboration of an existing argument is never necessary to make it a valid attacker).

Upon reception of a message $\mathsf{attack}(\mathbf{a} \rightharpoonup \mathbf{b})$, an agent σ with a portrayal instance $\mathcal{P}[\sigma]$ should ensure that arguments **a** and **b** are within $\mathcal{P}$ before invoking the verify_attack function:

Definition 5.20 *An agent σ updates its portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message* $\mathsf{attack}(\mathbf{a} \rightharpoonup \mathbf{b})$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{attack}(\mathbf{a} \rightharpoonup \mathbf{b})) & \wedge \\ \mathcal{P}[\sigma]_{\mathbf{b}} = \mathsf{insert}(\mathcal{P}[\sigma], \mathbf{b}) & \wedge \\ \mathcal{P}[\sigma]_{\mathbf{a}} = \mathsf{insert}(\mathcal{P}[\sigma]_{\mathbf{b}}, \mathbf{a}) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{insert\_attack}(\mathcal{P}[\sigma]_{\mathbf{a}}, \mathbf{a} \rightharpoonup \mathbf{b}) \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M)$ *is true if agent σ has received a message M from one of its peers in Σ of $\mathcal{P}$.*
- $\mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a})$ *returns an updated portrayal instance $\mathcal{P}[\sigma]$ such that $\mathbf{a} \in \mathcal{P}$, as specified in §5.4.2.*
- $\mathsf{insert\_attack}(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b})$ *returns an updated portrayal instance $\mathcal{P}[\sigma]$ in which $\mathbf{a} \rightharpoonup \mathbf{b}$ has been verified in $\mathcal{P}$.*

Function insert_attack ensures that a given attack relation exists within the portrayal and is valid:

$\mathcal{P}[\sigma]' = \mathsf{insert\_attack}(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b})$ — *Given an existing portrayal instance $\mathcal{P}[\sigma]$ and in response to an attack $\mathbf{a} \rightharpoonup \mathbf{b}$, a peer σ with theory context $\mathcal{C}$ should ensure that $\mathbf{a} \rightharpoonup \mathbf{b}$ according to $\mathcal{P}$ provided that:*

- $\mathbf{a}, \mathbf{b} \in \mathcal{P}$

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$

If these conditions are met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup)', \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$, *where* $\mathbf{a} \rightharpoonup \mathbf{b}$ *according to* $\mathcal{P}'$.
2. *If* $\mathbf{a} \rightharpoonup \mathbf{b}$ *is invalid with respect to* $\mathcal{C}$, *then invoke* $\mathsf{dismiss}(\mathcal{P}[\sigma]', \mathbf{a} \rightharpoonup \mathbf{b})$.

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

All insert_attack does in practice is ensure an attack is noted in whatever representation of the portrayal argument system an agent might use, and then check for validity from the executing agent's perspective. Note that the invocations of insert for the defending and then attacking arguments serve to ensure reconciliation, so we have no need to invoke reconcile within insert_attack.

5.4.5 Requesting Additional Elaboration Upon Arguments

The inquire operation is used by an agent to request that a peer elaborate upon a given argument if the agent has decided that insufficient information has been made available:

$\mathsf{inquire}(\mathcal{P}, \langle \Phi, \alpha \rangle, \mu)$ — *An agent* σ *can request that another agent* μ *elaborate upon an argument* $\mathbf{a}$ *within portrayal* $\mathcal{P}$ *if and only if:*

- *It is the case that* $\langle \Phi, \alpha \rangle \in \mathcal{P}$.
- *Agent* $\mu \in \Sigma$, *where* Σ *of* $\mathcal{P}$ *is the set of agents currently involved in the interaction for which* $\mathcal{P}$ *exists.*
- *Agent* σ *has an interest in the inference of one or more sentences* $\varphi \in \Phi$ *as described below.*

A message $\mathsf{inquire}(\mathbf{a})$ *can then be dispatched to agent* μ.

Any agent can request that any other agent elaborate if they can upon a given argument, as long as the argument exists within a portrayal. The motivation for such an event is at the discretion of the inquiring agent — of all parts of the portrayal mechanism, this is the one that confers the most freedom to agents, essentially allowing them to arbitrarily expand the portrayal space in order to gather more information. It should be noted that the inquire operation is not needed to portray an interaction — arguments are generally elaborated upon in order to be attacked, and if no admissible attacks exist against any

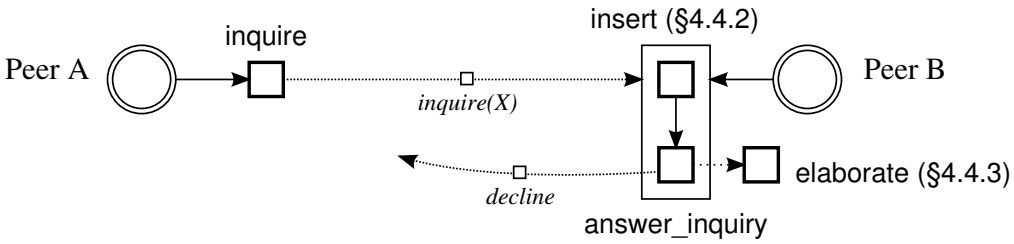


Figure 5.9: *Upon Peer A making a request of Peer B to elaborate upon a given argument, Peer B can do so provided that it is able, or else decline; if Peer B does not recognise the inquired-upon argument in the first place, it will automatically insert it into its portrayal instance anyway.*

elaboration of an argument, then it is not necessary to elaborate upon that argument to synchronise agent theories within the portrayal space. On the other hand, there are instances where an intelligent agent might simply want more information.

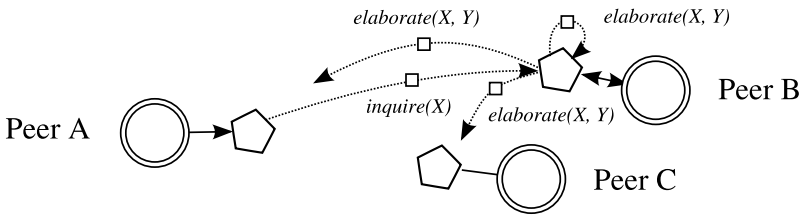
The most likely circumstance in which inquire might be invoked by an agent σ is where there exists a logical constraint imposed on interaction which is specifically imposed on σ in its enactment of a given role in the interaction, such that σ is the agent which must actually resolve a proposition α. If σ itself does not have a means to infer α as being true or false other than by assumption, then it might want to know how another agent is able to derive a claim made in the portrayal, even if it would not otherwise be elaborated upon (which might be the case if no other agent is able to undercut the claim).

Typically the agent asked to elaborate upon an argument **a** will be the agent which posited **a**, but this is not compulsory. It may be worth noting that the inquire operation, dependent on the temperament of the agents involved in an interaction, may not see common use — this is because agents can instead invite further discussion by elaborating upon arguments on their own initiative, often specifically to attack, and then allowing their peers to respond with alternative elaborations or counter-attacks as they wish.

Provided that all conditions are fulfilled, an agent will send a message $\mathsf{inquire}(\mathbf{a})$ to another agent σ. Upon reception of an inquiry, agent σ must first ensure that it has the subject of inquiry within its portrayal:

Subject of Inquiry is Present — If the subject of the inquiry is present within the portrayal, or if it is merely a potential argument for another argument already present in the portrayal, then the agent is free to elaborate upon it or otherwise:

Peer A inquires about an argument X to peer B:

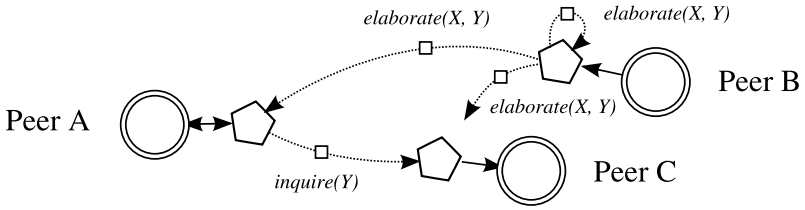


Assuming that argument X is present in B's portrayal instance, and B is able to construct a more concrete argument Y, then B can elaborate upon X to all agents handling the portrayal.

This is the default case.

Existing Arguments Need to be Expanded — If the given argument is already an elaboration of existing arguments in portrayal $\mathcal{P}$, then the recipient may as well replace those existing arguments with the elaboration as if that argument had just been inserted into the portrayal as per §5.4.2 or §5.4.3:

Given an elaboration Y by peer B, peer A seeks further elaboration by peer C:

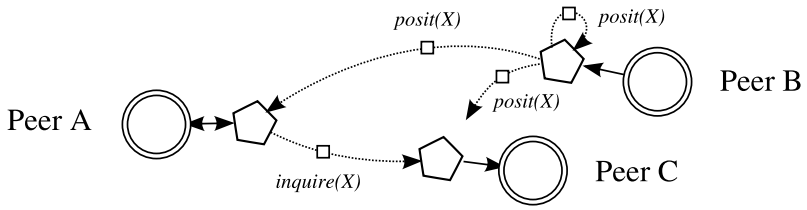


Even if elaboration Y is not yet present in C's portrayal instance, C can still elaborate on it further if it first expands the original argument X and is able to construct an even more concrete argument than Y.

This response may occur if an agent is asked to elaborate upon an argument it has not yet received itself (since generally agents are asked to elaborate upon their own arguments however, this case should be quite rare).

The Subject of Inquiry Needs to be Inserted — If the given argument is not present in the recipient's portrayal instance, and moreover if there exist no alternative arguments to elaborate in its stead, then the recipient may as well add the argument to its copy of the portrayal:

Peer A inquires about an argument X originally posited by peer B to peer C:



Even if argument X is not present in C's portrayal instance, C can still elaborate upon it if it has sufficient information in its theory context.

This response may occur for much the same reasons as for the previous case.

Therefore, upon reception of a message $\mathsf{inquire}(\mathbf{a})$, an agent σ with a portrayal instance $\mathcal{P}[\sigma]$ should insert the subject of inquiry if missing from its portrayal instance, and then see if it can elaborate upon it further:

Definition 5.21 *An agent σ updates its portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message* $\mathsf{inquire}(\mathbf{a})$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{inquire}(\mathbf{a}), \sigma_s) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a}) & \wedge \\ \mathsf{answer\_inquiry}(\mathcal{P}[\sigma]', \mathbf{a}, \sigma_s) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent σ has received a message M from peer σ_s in Σ of $\mathcal{P}$.*
- $\mathsf{insert}(\mathcal{P}[\sigma], \mathbf{a})$ *returns an updated portrayal instance $\mathcal{P}[\sigma]$ such that $\mathbf{a} \in \mathcal{P}$, as specified in §5.4.2.*
- $\mathsf{answer\_inquiry}(\mathcal{P}[\sigma], \mathbf{a}, \sigma_s)$ *either performs an elaboration of* **a** *or dispatches a message* $\mathsf{decline}(\mathbf{a})$ *to agent σ_s.*

Function answer_inquiry determines whether or not an agent can elaborate a given argument, and will then inform the inquirer as to its decision.

$\mathcal{P}[\sigma]' = \mathsf{answer\_inquiry}(\mathcal{P}[\sigma], \mathbf{a}, \sigma_s)$ — *Given an existing portrayal instance $\mathcal{P}[\sigma]$ and in response to an inquiry on an argument* **a** *from a peer σ_s, an agent σ with theory context $\mathcal{C}$ and an accepted extension $\mathcal{E}$ of $\mathcal{C}$ should elaborate upon* **a** *provided that:*

- $\mathbf{a} \in \mathcal{P}$.
- *There exists an argument* $\mathbf{b} \in \mathcal{E}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$.

- *There exists an elaboration* $\mathbf{c}$ *such that* $\mathbf{a} \sqsubset \mathbf{c} \sqsubseteq \mathbf{b}$.

If these conditions are met, then:

1. *Invoke* $\mathsf{elaborate}(\mathcal{P}[\sigma], \mathbf{a}, \mathbf{b})$.

If these conditions are not *met, then a message* decline *should be sent to agent* σ_s.

Outwith the bounds of the argument space of the portrayal, it is difficult to define precisely how far an agent should elaborate upon an argument subject to inquiry without knowledge of the specific logical construction of the argument in question. As such, it ultimately falls to the discretion of the elaborating agent. As a general rule, the elaboration should be non-trivial (such that it imparts genuinely new information), but should only be elaborated upon to the full extent of an agent's knowledge if the difference between the potential argument in the portrayal and 'full' argument in an agent's theory context is already quite small.

5.4.6 Observing Unarguable Propositions

The observe operation is used by agents to identify particular assertions used in arguments in a portrayal which they hold to be evidently true or false, regardless of the apparent naive admissibility of those assertions in the portrayal argument system:

$\mathsf{observe}(\mathcal{P}[\sigma], S)$ — *An agent* σ *with context theory* $\mathcal{C}$ *advocates the set of logical sentence S within a portrayal* $\mathcal{P}$ *if and only if:*

- *For each sentence* $\varphi \in S$*, it is the case that* $\Theta \vdash \varphi$*, where* Θ *is the theory core used within* $\mathcal{C}$.
- *For each sentence* $\varphi \in S$*, there exists an argument* $\langle \Phi, \alpha \rangle \in \mathcal{P}$ *such that* $\Phi \vdash \neg\varphi$ *according to* $\mathcal{C}$.
- *For each sentence* $\varphi \in S$*, it is the case that* $\varphi \notin \mathsf{obs}(\sigma)$ *according to* $\mathcal{C}$*, where* $\mathsf{obs}(\sigma)$ *is the set of sentences already advocated by* σ *in* $\mathcal{P}[\sigma]$.

A message $\mathsf{observe}(S)$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

The observe operation is used by an agent to inform its peers about assertions which logically follow from arguments already in the portrayal which the agent is certain are

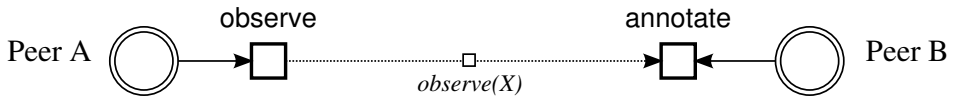


Figure 5.10: *Upon Peer A asserting an observation of the environment, each agent should update its record of Peer A's observations.*

true or false.[6] This certainty arises from outside of the argumentation framework in which these assertions are being evaluated (i.e. the truth status of these assertions has been directly deduced by introspection or observation). Basically, the agent is stating that it does not consider to be admissible any set of arguments which contradict those assertions, such arguments being 'clearly false' given the advocate's knowledge.

Observations in a portrayal can be used to dismiss arguments as per Definition 3.15. Agents are free to take into account or ignore the observations of their peers when evaluating the portrayal — generally it is better to try and interpret arguments given all peers' observations, but since it cannot be ensured that the aggregation of such will be consistent, an agent must sometimes treat their peers' opinions as merely a heuristic for choosing between interpretations (see §5.1.5).

The decision by an agent σ with theory context $\mathcal{C}$ to enact the advocate operation is generally motivated by one of two occurrences:

- An agent has posited (or elaborated upon) a new argument $\langle \Phi, \alpha \rangle$ in portrayal $\mathcal{P}$ such that there exists at least one sentence φ for which $\Phi \vdash \varphi$ and $\Theta \vdash \neg\varphi$ (where Θ is the theory core of $\mathcal{C}$), and $\varphi \notin \mathsf{obs}(\sigma)$ or $\neg\varphi \notin \mathsf{obs}(\sigma)$ respectively (where obs is the advocacy function of $\mathcal{P}$).

- There has been a change in the environment, motivating a change in Θ of $\mathcal{C}$ such that there exists at least one sentence φ entailed by some admissible extension in $\mathcal{P}$ for which $\Theta \vdash \neg\varphi$, and $\neg\varphi \notin \mathsf{obs}(\sigma)$ respectively.

If an agent wishes to advocate something which directly contradicts a prior observation, then it can do so — peers will automatically remove the original assertion when they update their instances of the portrayal advocacy function. If an agent merely wishes to retract a prior observation (because the assertion in question has merely become uncertain rather than empirically untrue), then it should invoke the unobserve operation (§5.4.7).

[6] For a given standard of certainty — recall the discussion in §5.1.5.

Note that we only concern ourselves with positing observations which contradict rather than support suppositions made in arguments within a portrayal. This is for much the same reason as given in §3.2.3 — if an assertion supported by an observation is attacked, then the observation can be used to dismiss that attack; if an assertion supported by an observation is not attacked, then it will be assumed as part of an interpretation of the portrayal argument system anyway.

Given the decision to advocate a particular observation, a message $\mathsf{observe}(S)$ will be sent to all agents possessing instances of portrayal $\mathcal{P}$, at which point those agents will update their observation functions with respect to the observer:

Definition 5.22 *An agent σ updates its portrayal instance $\mathcal{P}[\sigma]$ upon reception of a message* $\mathsf{observe}(S)$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{l} \mathsf{received}(\sigma, \mathsf{observe}(S), \sigma_s) \quad \wedge \\ \mathcal{P}[\sigma]' = \mathsf{annotate}(\mathcal{P}[\sigma], S, \sigma_s) \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent σ has received a message M from peer σ_s in Σ of $\mathcal{P}$.*
- $\mathsf{annotate}(\mathcal{P}[\sigma], S, \sigma_s)$ *updates portrayal instance $\mathcal{P}[\sigma]$ such that the sentences in set S are recorded as having been observed by peer σ_s.*

Function annotate updates the observation function obs for the given agent:

$\mathcal{P}[\sigma]' = \mathsf{annotate}(\mathcal{P}[\sigma], S, \sigma_s)$ — *Given an existing portrayal instance $\mathcal{P}[\sigma]$ and in response to the observation of the logical sentence set S by a peer σ_s, an agent σ should update* $\mathsf{obs}(\sigma_s)$ *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$.
- $R = \{\varphi \in \mathsf{obs}(\sigma_s) \mid \neg\varphi \in S\}$

If these conditions are met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}', \mathsf{acc})$ *where:*
   (a) *If $\varphi \in S$, then $\varphi \in \mathsf{obs}'(\sigma_s)$.*
   (b) *If $\varphi \in \mathsf{obs}(\sigma_s)$ and $\varphi \in R$, then $\varphi \notin \mathsf{obs}'(\sigma_s)$.*
   (c) *Otherwise, if $\varphi \in \mathsf{obs}(\mu)$ for any agent $\mu \in \Sigma$ of $\mathcal{P}$, then $\varphi \in \mathsf{obs}'(\mu)$.*

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

Note that whilst annotate automatically removes any directly contradicted past observations when updating the observation function for efficiency, it does not do full consistency checking — it is the responsibility of the observer to determine that its observations are internally consistent. If an agent is unable to provide consistent observations, then it is unlikely that peers will choose to include its observations within their theory cores; on the other hand, if the source of observations is sound, no inconsistencies should arise in the first place provided that observations are kept updated.

Finally, agents will record observations made even if there is no apparent argument which the observation can be used to dismiss — this is to account for the circumstance in which a third agent receives the observation made by one agent prior to the argument made by another agent which motivated that observation.

5.4.7 Withdrawing Prior Observations

The unobserve operation is used by agents to retract advocacy of observations used in arguments in a portrayal. This is done in response to a change in the state of the environment removing a previously observed percept:

unobserve$(\mathcal{P}[\sigma], R)$ — *An agent* σ *with theory context* $\mathcal{C}$ *disavows the set of logical sentences* R *within a portrayal* $\mathcal{P}$ *if and only if:*

- *For each sentence* $\varphi \in R$*, it is the case that* $\varphi \in \mathsf{obs}(\sigma)$ *according to* $\mathcal{C}$*, where* $\mathsf{obs}(\sigma)$ *is the set of sentences already advocated by* σ *in* $\mathcal{P}[\sigma]$.
- *For each sentence* $\varphi \in R$*, it is the case that* $\Theta \nvdash \varphi$*, where* Θ *is the theory core used within* $\mathcal{C}$.

A message unobserve(R) *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

The unobserve operation is used by an agent σ to essentially retract its advocacy of an assertion. This operation is usually motivated by a change in the environment causing a change in the theory core Θ of a theory content $\mathcal{C}$ such that there exists at least one sentence $\varphi \in \mathsf{obs}(\sigma)$ (where obs is the advocacy function of portrayal $\mathcal{P}$) for which $\Theta \nvdash \varphi$ (i.e. assertion φ has become false or unknown). There is no need to unobserve any assertion for which its negation is due to be observed however, because the observe operation will cause any directly contradicted prior observations to be overwritten (see

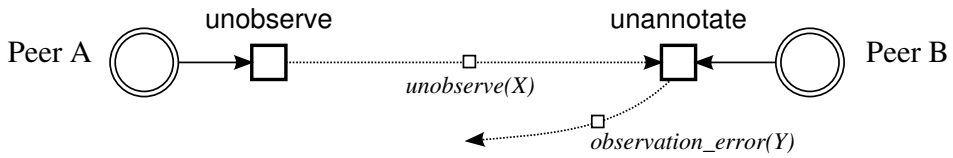


Figure 5.11: *Upon the retraction of observations by Peer A, each agent should update its record of Peer A's observations; if a retraction appears to proceed an assertion, then peers can also throw an observation error.*

§5.4.6); therefore, unobserve is mainly reserved for previously observable assertions which become unknowable.

Given the decision to disavow a particular observation, a message $\mathsf{unobserve}(R)$ will be sent to all agents possessing instances of portrayal $\mathcal{P}$, at which point those agents will update their observation functions with respect to the observer:

Definition 5.23 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* $\mathsf{unobserve}(R)$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{unobserve}(R), \sigma_s) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{unannotate}(\mathcal{P}[\sigma], R, \sigma_s) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent* σ *has received a message* M *from peer* σ_s *in* Σ *of* $\mathcal{P}$*.*
- $\mathsf{unannotate}(\mathcal{P}[\sigma], R, \sigma_s)$ *updates portrayal instance* $\mathcal{P}[\sigma]$ *such that the sentences in set* R *are no longer recorded as being observed by peer* σ_s*.*

Function unannotate updates the observation function obs for the given agent:

$\mathcal{P}[\sigma]' = \mathsf{annotate}(\mathcal{P}[\sigma], R, \sigma_s)$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the un-observation of the logical sentence set* R *by a peer* σ_s*, an agent* σ *should update* $\mathsf{obs}(\sigma_s)$ *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$
- $R \subseteq \mathsf{obs}(\sigma_s)$.

If these conditions are met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}', \mathsf{acc})$ *where:*

(a) If $\varphi \in \mathsf{obs}(\sigma_s)$ *and* $\varphi \in R$*, then* $\varphi \notin \mathsf{obs}'(\sigma_s)$.

(b) Otherwise, if $\varphi \in \mathsf{obs}(\mu)$ *for any agent* $\mu \in \Sigma$ *of* $\mathcal{P}$*, then* $\varphi \in \mathsf{obs}'(\mu)$.

If $R \nsubseteq \mathsf{obs}(\sigma_s)$*, then:*

1. $\mathcal{P}[\sigma]' = \mathsf{annotate}(\mathcal{P}[\sigma], (R \cap \mathsf{obs}(\sigma_s)), \sigma_s)$.
2. *Dispatch a message* $\mathsf{observation\_error}(R/\mathsf{obs}(\sigma_s))$ *to* σ_s.

Otherwise, $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

If an agent is told to withdraw an observation on the part of a peer *before* the agent receives notification of the original observation, then there may have been a disordering of messages. In order to avoid ending up with an incorrect record of observations, an agent can send an observation_error, which informs a peer that their observations do not appear to have been received in order and returns the portion of sentence set R which is in doubt. In response to such a message, the peer should either:

- Repeat the unobserve operation for the portion of R in doubt, if the peer believes that there is a lagging observe message in transit such that by the time the second unobserve message is received, the missing observations will have been recorded by the recipient.

- Do nothing, if the peer believes that the lagging observe message has been lost (in which case there is nothing to correct).

Because we are considering an asynchronous system in which a true state of common knowledge cannot be reached [Halpern and Moses, 1990], we cannot ensure absolutely that the set of observations in a portrayal instance is perfectly correct. We are able to minimise this problem for arguments by making the portrayal primarily additive (we never simply remove arguments without an elaboration to replace them), such that arguments can be received out-of-order with little difficulty (though it is still possible for arguments to become lost in the ether) — however we cannot do this for observations, which genuinely can be retracted as often as they are added to. As a general heuristic, we would expect that an agent would repeat an unobserve message once in response to a observation_error, and then to ignore further observation_error messages for the same set of logical sentences until the theory core of the agent changes again.

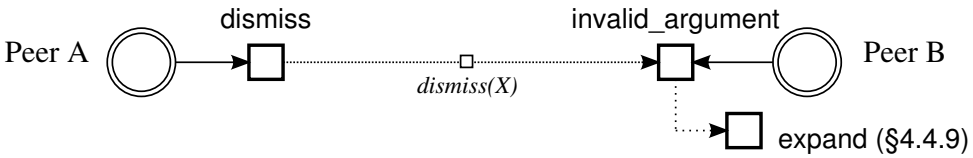


Figure 5.12: *Upon the dismissal of an argument or attack by Peer A, each agent should update its record of invalid arguments / attacks and, if able, should provide a clarification of the dismissed argument or a clarification of the attacking argument in a dismissed attack.*

5.4.8 Dismissing Invalid Arguments and Attacks

The dismiss operation is used by agents to identify arguments and attacks which are not valid with respect to their theory contexts. This operation is only used for *public* dismissals due to invalid arguments or attacks — it is *not* used by agents to privately dismiss arguments which have been observed to be false, but which are still valid inferences given the assumption of the argument's support (operation observe handles this; §5.4.6). The operation has two variants. For invalid arguments:

$\mathsf{dismiss}(\mathcal{P}[\sigma], \mathbf{a})$ — *An agent* σ *with theory context* $\mathcal{C}$ *dismisses an argument* $\mathbf{a}$ *within portrayal* $\mathcal{P}$ *if and only if:*

- *There exists an argument* $\mathbf{a} \in \mathcal{P}$.
- $\mathbf{a}$ *is invalid with respect to* $\mathcal{C}$ *as per Definition 4.11.*
- $\mathbf{a} \notin \mathsf{inv}_{\mathcal{A}}(\sigma)$, *where* $\mathsf{inv}_{\mathcal{A}}$ *is the invalid argument function of* $\mathcal{P}[\sigma]$.

A message $\mathsf{dismiss}(\mathbf{a})$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

An argument is considered for validity upon being inserted into the portrayal (see §5.4.2). An argument declared invalid indicates a peer which is unable to infer the argument's claim from its support. In turn, this indicates that an agent has presumed a certain proposition or rule as being known to all agents (and believed to be always true by all agents) such that it need not be explicitly mentioned in arguments, and that presumption is false. Equivalently, for invalid attacks:

$\mathsf{dismiss}(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b})$ — *An agent* σ *with theory context* $\mathcal{C}$ *dismisses an attack* $\mathbf{a} \rightharpoonup \mathbf{b}$ *within* $\mathcal{P}$ *if and only if:*

- $\mathbf{a} \rightharpoonup \mathbf{b}$ *according to* $\mathcal{P}$.

- $\mathbf{a} \rightharpoonup \mathbf{b}$ *is invalid with respect to* $\mathcal{C}$ *as per Definition 4.12.*
- $(\mathbf{ab}) \notin \mathsf{inv}_{\rightharpoonup}(\sigma)$*, where* $\mathsf{inv}_{\rightharpoonup}$ *is the invalid attack function of* $\mathcal{P}[\sigma]$.

A message $\mathsf{dismiss}(\mathbf{a} \rightharpoonup \mathbf{b})$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

An attack is considered for validity upon being made explicit within the portrayal (see §5.4.4). An attack declared invalid indicates a peer which is unable to determine how the claim of the attacking argument undercuts or rebuts the target argument. This also indicates a presumption on the part of an agent, in this case that two assertions are mutually exclusive, which is not shared by the peer.

In either case, there should be scope to repair the invalid argument or attack by making the claim or contradiction clearer. We rely on a base-line of deductive reasoning; if the claim of an argument can be inferred purely by deduction on the supporting assumptions used by that argument, then it should be valid to all peers. Likewise, if an attacking argument actually directly claims the negation of a proposition deducible from the target argument, then the attack should also be valid to all peers.

Given the decision to dismiss an argument, a message $\mathsf{dismiss}(\mathbf{a})$ is sent to all agents possessing instances of portrayal $\mathcal{P}$. Upon reception of such a message, an agent σ will update its invalid argument function with respect to the dismissing peer:

Definition 5.24 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* $\mathsf{dismiss}(\mathbf{a})$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{dismiss}(\mathbf{a}), \sigma_s) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{invalid\_argument}(\mathcal{P}[\sigma], \mathbf{a}, \sigma_s) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent* σ *has received a message* M *from peer* σ_s *in* Σ *of* $\mathcal{P}$.
- $\mathsf{invalid\_argument}(\mathcal{P}[\sigma], \mathbf{a}, \sigma_s)$ *updates portrayal instance* $\mathcal{P}[\sigma]$ *such that argument* $\mathbf{a}$ *is recorded as having been dismissed by peer* σ_s.

Function invalid_argument updates the invalid argument function $\mathsf{inv}_{\mathcal{A}}$ for the given agent:

$\mathcal{P}[\sigma]' = \mathsf{invalid\_argument}(\mathcal{P}[\sigma], \mathbf{a}, \sigma_s)$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the dismissal of an argument* $\mathbf{a}$ *by a peer* σ_s*, an agent* σ *with theory context* $\mathcal{C}$ *and an accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$ *should update* $\mathsf{inv}_{\mathcal{A}}(\sigma_s)$ *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$

If this condition is met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}'_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *where:*
   (a) $\mathbf{a} \in \mathsf{inv}_{\mathcal{A}}(\sigma_s)$.
   (b) Otherwise, if $\mathbf{b} \in \mathsf{inv}_{\mathcal{A}}(\mu)$ *for any agent* $\mu \in \Sigma$ *of* $\mathcal{P}$*, then* $\mathbf{b} \in \mathsf{inv}'_{\mathcal{A}}(\mu)$.
2. *If there exists an argument* $\mathbf{b} \in \mathcal{E}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$*, then invoke* $\mathsf{expand}(\mathcal{P}[\sigma], \mathbf{a}, \langle \Phi, \alpha \rangle)$*, where:*
   (a) $\langle \Phi, \alpha \rangle \in \Delta$ *(where* Δ *is the argument space of* $\mathcal{P}$*).*
   (b) $\mathbf{a} \sqsubseteq \langle \Phi, \alpha \rangle \sqsubseteq \mathbf{b}$.
   (c) $\Phi \vdash \alpha$ *in* any *deductive framework* $(\mathcal{L}, \vdash)$ *(i.e.* $\langle \Phi, \alpha \rangle$ *is a replacement for* $\mathbf{a}$ *which presumes nothing, but is still minimal with respect to the portrayal space of* $\mathcal{P}$*).*

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

Given the decision to dismiss an attack, a message $\mathsf{dismiss}(\mathbf{a} \rightharpoonup \mathbf{b})$ is sent to all agents possessing instances of portrayal $\mathcal{P}$. Upon reception of such a message, an agent σ will update its invalid argument function with respect to the dismissing peer:

Definition 5.25 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* $\mathsf{dismiss}(\mathbf{a} \rightharpoonup \mathbf{b})$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{dismiss}(\mathbf{a} \rightharpoonup \mathbf{b}), \sigma_s) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{invalid\_attack}(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b}, \sigma_s) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent* σ *has received a message* M *from peer* σ_s *in* Σ *of* $\mathcal{P}$.
- $\mathsf{invalid\_attack}(\mathcal{P}[\sigma], \mathbf{a} \rightharpoonup \mathbf{b}, \sigma_s)$ *updates portrayal instance* $\mathcal{P}[\sigma]$ *such that attack* $\mathbf{a} \rightharpoonup \mathbf{b}$ *is recorded as having been dismissed by peer* σ_s.

Function invalid_attack updates the invalid argument function $\mathsf{inv}_{\rightharpoonup}$ for the given agent:

$\mathcal{P}[\sigma]' = \mathsf{invalid\_attack}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle, \sigma_s)$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the dismissal of an attack* $\langle \Phi, \alpha \rangle \rightharpoonup \langle \Psi, \beta \rangle$ *by a peer* σ_s*, an agent* σ *with theory context* $\mathcal{C}$ *and an accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$ *should update* $\mathsf{inv}_{\rightharpoonup}(\sigma_s)$ *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$

If this condition is met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}'_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ *where:*
   (a) $(\langle \Phi, \alpha \rangle, \langle \Psi, \beta \rangle) \in \mathsf{inv}_{\mathcal{A}}(\sigma_s)$.
   (b) Otherwise, if $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoonup}(\mu)$ *for any agent* $\mu \in \Sigma$ *of* $\mathcal{P}$*, then* $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}'_{\rightharpoonup}(\mu)$.
2. *If there is an argument* $\mathbf{a} \in \mathcal{E}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{a}$*, invoke* $\mathsf{expand}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle, \langle \Phi', \neg\gamma \rangle)$*, where:*
   (a) $\langle \Phi', \neg\gamma \rangle \in \Delta$ *(where* Δ *is the argument space of* $\mathcal{P}$*).*
   (b) $\langle \Phi', \alpha \rangle \sqsubseteq \mathbf{a}$.
   (c) $\langle \Phi' \vdash \alpha \rangle$ *and* $\Psi \vdash \gamma$ *in* any *deductive framework* $(\mathcal{L}, \vdash)$ *(i.e.* $\langle \Phi', \neg\gamma \rangle$ *is a replacement for* $\mathbf{a}$ *which directly contradicts* $\langle \Psi, \beta \rangle$ *and which is still minimal with respect to the portrayal space of* $\mathcal{P}$*).*

If these conditions are not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

For both invalid_argument and invalid_attack, we do not care if an agent receives notification of an invalid argument or attack before notification of the argument or attack itself, since we can still record the invalid argument or attack even without it being present in the argument system — such an occurrence does not have any effect on the interpretation of the argument system itself anyway.

5.4.9 Validating Invalid Arguments and Attacks

The expand operation is used to replace invalid arguments, or the attacking argument of an invalid attack, with an expanded argument which more explicitly states its contribution to a portrayal argument system:

$\mathsf{expand}(\mathcal{P}[\sigma], \mathbf{a}, \mathbf{b})$ — *An agent* σ *with context theory* $\mathcal{C}$ *and an accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$ *expands upon an argument* $\mathbf{a}$*, replacing it with an argument* $\mathbf{b}$ *within a portrayal* $\mathcal{P}$ *if:*

- $\mathbf{a} \in \mathcal{P}$ *and for at least one agent* $\mu \in \Sigma$ *of* $\mathcal{P}$*, it is the case that* $\mathbf{a} \in \mathsf{inv}_{\mathcal{A}}(\mu)$*, where* $\mathsf{inv}_{\mathcal{A}}$ *is the invalid argument function of portrayal instance* $\mathcal{P}[\sigma]$.
- *There exists an argument* $\mathbf{c} \in \mathcal{E}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{c}$.

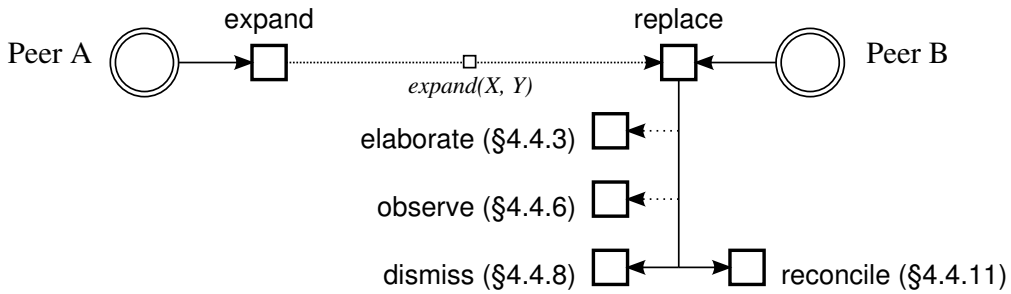


Figure 5.13: *Upon one of its arguments or attacks being declared invalid, Peer A can expand upon arguments in order to make their consequence clearer, at which point its peers can re-evaluate replacement arguments.*

- $\mathbf{b} \in \Delta$ *and* $\mathbf{a} \sqsubseteq \mathbf{b} \sqsubseteq \mathbf{c}$.
- $\mathbf{b} = \langle \Phi, \alpha \rangle$ *such that* $\Phi \vdash \alpha$ *in any deductive framework* $(\mathcal{L}, \vdash)$.

Or if:

- $\mathbf{a} \in \mathcal{P}$ *and for at least one agent* $\mu \in \Sigma$ *of* $\mathcal{P}$*, it is the case that* $(\mathbf{a}, \langle \Psi, \beta \rangle) \in \mathsf{inv}_{\rightharpoonup}(\mu)$*, where* $\mathsf{inv}_{\rightharpoonup}$ *is the invalid attack function of portrayal instance* $\mathcal{P}[\sigma]$.
- *There exists an argument* $\mathbf{c} \in \mathcal{L}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{c}$.
- $\mathbf{b} \in \Delta$ *and* $\mathbf{b} \sqsubseteq \mathbf{c}$.
- $\mathbf{a} = \langle \Phi, \alpha \rangle$ *and* $\mathbf{b} = \langle \Phi', \neg\gamma \rangle$ *such that* $\Phi' \vdash \alpha$ *and* $\Psi \vdash \gamma$ *in any deductive framework* $(\mathcal{L}, \vdash)$.

In either case, a message $\mathsf{expand}(\mathbf{a}, \mathbf{b})$ *is dispatched to all agents in* Σ *of* $\mathcal{P}$.

'Expansion' of arguments in this case is different from elaboration of arguments insomuch as instead of revealing the provenance of assumptions, expansion actually makes explicit parts of the support for the argument which were subsumed by the logical framework with which the original argument was created. In this case there are two variants of expansion: one which reinforces the support for an argument by making it clearer how the claim is inferred from the conclusion; and one in which the claim itself is replaced to more clearly demonstrate how it attacks another argument.

Given the decision to expand a given argument, a message $\mathsf{expand}(\mathbf{a}, \mathbf{b})$ will be sent to all agents possessing instances of $\mathcal{P}$, at which point those agents will replace the argument **a** with the expanded argument **b**:

Definition 5.26 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* $\mathsf{expand}(\mathbf{a},\mathbf{b})$*:*

$$\mathsf{update}(\sigma,\mathcal{P}[\sigma],\mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma,\mathsf{expand}(\mathbf{a},\mathbf{b})) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{replace}(\mathcal{P}[\sigma],\mathbf{b}) & \end{array}\right)$$

Where:

- $\mathsf{received}(\sigma,M)$ *is true if agent* σ *has received a message* M *from one of its peers in* Σ *of* $\mathcal{P}$*.*
- $\mathsf{replace}(\mathcal{P}[\sigma],\mathbf{a},\mathbf{b})$ *updates portrayal instance* $\mathcal{P}[\sigma]$ *such that argument* $\mathbf{a}$ *is replaced by argument* $\mathbf{b}$*.*

Function replace simply replaces an argument with another. replace is identical to insert (§5.4.2), but with a very minor difference:

$\mathcal{P}[\sigma]' = \mathsf{replace}(\mathcal{P}[\sigma],\mathbf{a})$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the expansion of an argument* $\mathbf{a}$*, a peer* σ *with theory context* $\mathcal{C}$ *should insert* $\mathbf{a}$ *into* $\mathcal{P}$ *provided that:*

- *There exists no argument* $\mathbf{a} \in \mathcal{P}$ *such that* $\langle\Phi,\alpha\rangle \sqsubset \mathbf{a}$*.*
- *S is the set of all arguments* $\mathbf{b} \in \mathcal{P}$ *such that* $\mathbf{b} \sqsubseteq \langle\Phi,\alpha\rangle$*.*
- $\mathcal{P}[\sigma] = (\Sigma,\Upsilon,(\mathcal{A},\rightharpoonup),\mathsf{inv}_{\mathcal{A}},\mathsf{inv}_{\rightharpoonup},\mathsf{obs},\mathsf{acc})$*.*

If these conditions are met, then do as for function insert*; otherwise,* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$*.*

The difference is that replace can replace arguments which, to an agent, are functionally identical to the given argument — this is because to an agent which considers valid an argument invalid to a peer, the expansion of an argument does not actually expand the inferences which can be made from that argument. Otherwise, replace possesses the same qualities as insert, including the ability to handle the replacement of an argument which has not yet arrived due to asynchronous messaging.

5.4.10 Asserting Acceptance of Portrayal Arguments

The accept operation is used by agents to identify the set of arguments in a portrayal which reflect their actual underlying beliefs as described by their theory contexts:

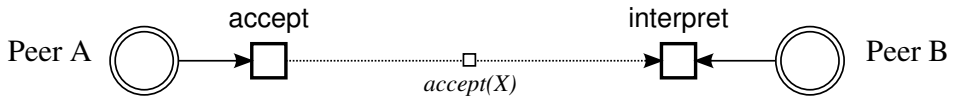


Figure 5.14: *Upon the declaration of accepted arguments by Peer A, each agent should update its record of Peer A's accepted extension of the portrayal argument system.*

$\mathsf{accept}(\mathcal{P}[\sigma], S)$ — *An agent* σ *with context theory* $\mathcal{C}$ *and an accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$ *declares the acceptance of a set of arguments S within a portrayal* $\mathcal{P}$ *if and only if:*

- *For every argument* $\mathbf{a} \in S$*, it is the case that* $\mathbf{a} \in \mathcal{P}$.
- *Argument set S is a potential restriction of* $\mathcal{E}$ *into the argument space* Δ *of* $\mathcal{P}$.
- *Argument set* $S \neq \mathsf{acc}(\sigma)$ *of* $\mathcal{P}$.

A message $\mathsf{accept}(S)$ *is then dispatched to all agents in* Σ *of* $\mathcal{P}$.

The accept operation is used by an agent to indicate the set of arguments it currently accepts for the purpose of making decisions in the interaction to which the portrayal is attached, and by extension the assumptions which it has chosen to make. Thus its peers can compare the agent's accepted argument set with their own, and determine the current efficacy of the arguments made so far.

An agent σ will declare its acceptance of arguments upon portrayal conception (§5.2.2) or after reconciling arguments in a portrayal with its theory context (§5.4.11) if the output of $\mathsf{acc}(\sigma)$ of its portrayal instance $\mathcal{P}[\sigma]$ is not a potential restriction of its (revised) beliefs. Thus, there exists the possibility of re-declaration of acceptance every time new arguments are added to the portrayal, or if there is a change in an agent's theory context from an external source.

Given the decision to declare acceptance of a given set of arguments *S*, a message $\mathsf{accept}(S)$ is sent to all agents possessing instances of $\mathcal{P}$, leading the recipients to immediately revise their acceptance functions accordingly:

Definition 5.27 *An agent* σ *updates its portrayal instance* $\mathcal{P}[\sigma]$ *upon reception of a message* $\mathsf{accept}(S)$*:*

$$\mathsf{update}(\sigma, \mathcal{P}[\sigma], \mathcal{P}[\sigma]') \quad \leftarrow \quad \left(\begin{array}{ll} \mathsf{received}(\sigma, \mathsf{accept}(S), \sigma_s) & \wedge \\ \mathcal{P}[\sigma]' = \mathsf{interpret}(\mathcal{P}[\sigma], S, \sigma_s) & \end{array} \right)$$

Where:

- $\mathsf{received}(\sigma, M, \sigma_s)$ *is true if agent* σ *has received a message* M *from peer* σ_s *in* Σ *of* $\mathcal{P}$.
- $\mathsf{interpret}(\mathcal{P}[\sigma], S, \sigma_s)$ *updates portrayal instance* $\mathcal{P}[\sigma]$ *such that the arguments in set* S *are recorded as having been accepted by peer* σ_s.

Function interpret updates the observation function acc for the given agent:

$\mathcal{P}[\sigma]' = \mathsf{interpret}(\mathcal{P}[\sigma], S, \sigma_s)$ — *Given an existing portrayal instance* $\mathcal{P}[\sigma]$ *and in response to the acceptance of the argument set* S *by a peer* σ_s*, an agent* σ *should update* $\mathsf{acc}(\sigma_s)$ *provided that:*

- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$.

If this condition is met, then:

1. $\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc}')$ *where:*
   (a) $\mathsf{acc}'(\sigma) = S$.
   (b) Otherwise, $\mathsf{acc}'(\mu) = \mathsf{acc}(\mu)$ *for all other agents* $\mu \in \Sigma$ *of* $\mathcal{P}$.

If this condition is not *met, then* $\mathcal{P}[\sigma]' = \mathcal{P}[\sigma]$.

If an agent declares acceptance of an argument which is not yet present in the portrayal, then the acceptance of the argument is noted, but the argument itself is *not* added to the portrayal's system of arguments — if there is a delayed argument to come, then it will be added when that happens. Note that the acceptance function is *not* automatically updated when constituent arguments are elaborated upon, because the elaboration might not be one accepted by a given agent in its theory context. There also exists the case in which an elaboration is declared invalid, whereupon the potential argument might be acceptable, but not the elaboration (see §5.1.4).

5.4.11 Reconciliation

The reconcile procedure precipitates the reconciliation of an agent's portrayal instance with its theory context as per Definition 4.10. In particular, the reconcile function ensures that all arguments in a portrayal instance are present in some form within an agent's theory context before checking to see if it knows any suitable counter-arguments:

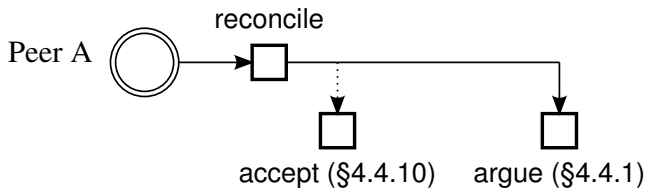


Figure 5.15: *Upon the addition of new arguments to the portrayal, each agent must reconcile those arguments with its own beliefs, revising accepted arguments if necessary and searching for any counter-arguments.*

reconcile$(\mathcal{P}[\sigma], S)$ — *Given a portrayal instance* $\mathcal{P}[\sigma]$ *containing a set of unverified arguments S, an agent* σ *in context* $\mathcal{C}$ *should reconcile* $\mathcal{C}$ *with* $\mathcal{P}$*, where:*

- *N is the set of all arguments* $\langle \Phi, \alpha \rangle \in S$ *valid according to* $\mathcal{C}$ *for which there exists no argument* $\mathbf{a} \in \mathcal{C}$ *such that* $\langle \Phi, \alpha \rangle \sqsubseteq \mathbf{a}$.

If $N \neq \emptyset$*, then* $\mathcal{C}$ *should be revised:*

1. *R is the set of all arguments* $\mathbf{b} \in \mathcal{C}$ *such that* $\mathbf{b} \sqsubset \langle \Phi, \alpha \rangle$ *for some argument* $\langle \Phi, \alpha \rangle \in N$.
2. *Extend the argument space* Δ_C *of* $\mathcal{C}$ *to include the set* $\bigcup_{\langle \Phi, \alpha \rangle \in N} \Phi$ *as per Definition 4.4.*
3. *Replace arguments* $\mathcal{A}$ *in* $\mathcal{C}$ *with the potential expansion of* $(\mathcal{A}/R) \cup N$ *into* Δ_C *as per Definition 4.3 and recompute the accepted extension* $\mathcal{E}_A$ *of* $\mathcal{C}$.
4. *If* acc(σ) *is not a potential restriction of* $\mathcal{E}_A$ *into the argument space* Δ_P *of* $\mathcal{P}$ *(where* $\mathcal{E}_A$ *is the accepted extension of* $\mathcal{C}$*), then invoke* accept$(\mathcal{P}[\sigma], S)$*, where S is the subset of the potential restriction* $\mathcal{E}'$ *of* $\mathcal{E}_A$ *into* Δ_P *such that if* $\varphi \in S$*, then* $\varphi \in \mathcal{P}$.

Regardless, invoke argue$(\mathcal{P}[\sigma])$.

The reconcile procedure verifies whether or not any new arguments added to a portrayal have been considered privately by peers. If an agent encounters an argument not represented within its theory context, then it should extend its hypothesis space in order to account for the premises used by the argument, and then perform a potential expansion of its private argument system so that the new argument is duly represented and its influence on existing arguments is duly factored into its reasoning. If it then finds that

it does not accept that argument, then it can prepare counter-arguments; if it concedes the argument, then it should formally accept the argument within the portrayal.

Together with the argue predicate, reconcile defines the process by which it can be ensured that a portrayal is reconciled with an agent's theory context as per Definition 4.10. As stated in §4.2 and proven by Proof 4.6, if all agents can reconcile a shared system of arguments with their theory contexts, then the beliefs of those agents will be synchronised within the argument space of that argument system (assuming that all invalid arguments and attacks are removed, and that no accepted extension of a theory context is contradicted by any peer's observations). If we can demonstrate that the portrayal mechanism as described in this chapter will reconcile a portrayal with every theory context used by an agent with an instance of the portrayal, then we can show that agents beliefs will be synchronised by the act of constructing a portrayal.

In order to clarify what we need to demonstrate, we first reiterate Definition 4.10 with respect to interaction portrayals:

Definition 5.28 *A portrayal instance $\mathcal{P}[\sigma]$ with an argument space Δ is* **reconciled** *with a theory context $\mathcal{C}$ of agent σ if and only if:*

- *Every complete extension of $\mathcal{P}[\sigma]$ given the theory core Θ of $\mathcal{C}$ is a potential restriction into Δ of an admissible extension of $\mathcal{C}$.*
- *There exists a potential restriction $\mathcal{E}'$ of the accepted extension $\mathcal{E}_A$ of $\mathcal{C}$ into Δ such that $\mathcal{E}'$ is an admissible extension of $\mathcal{P}$ given Θ.*

We will now demonstrate that pending additional updates to a portrayal $\mathcal{P}$ from other peers, and other external changes to agent σ's theory context $\mathcal{C}$, an agent σ which enacts predicate reconcile followed by argue will find that its (updated) portrayal instance $\mathcal{P}[\sigma]$ is reconciled with its (revised) context $\mathcal{C}$:

Theorem 5.2 *If, for any agent $\sigma \in \Sigma$ of $\mathcal{P}$, its portrayal instance $\mathcal{P}[\sigma]$ is not reconciled with its theory context $\mathcal{C}$, then the portrayal mechanism will continue to invoke the* reconcile *and* argue *procedures until reconciliation is attained.*

Proof 5.2 *Assume that the portrayal instance $\mathcal{P}[\sigma]$ of a portrayal $\mathcal{P}$ with argument space Δ held by an agent σ is* not *reconciled with the theory context $\mathcal{C}$ of σ. This means that either:*

1. *There exists a complete extension $\mathcal{E}'$ of $\mathcal{P}[\sigma]$ given theory core Θ of $\mathcal{C}$ which is not a potential restriction into Δ of an admissible extension $\mathcal{E}$ of $\mathcal{C}$.*

2. *There does not exist a potential restriction $\mathcal{E}'$ of the accepted extension $\mathcal{E}_A$ of $\mathcal{C}$ into Δ such that $\mathcal{E}'$ is an admissible extension of $\mathcal{P}$ given Θ.*

Consider each case in turn:

1. *Consider the first case. By Definition 4.2, we can infer that there is no admissible extension $\mathcal{E}$ of $\mathcal{C}$ where:*

   (a) *For each argument $\mathbf{a} \in \mathcal{E}'$, it is the case that $\mathbf{a} \in \Delta$ of $\mathcal{P}$ and $\mathbf{a} \sqsubseteq \mathbf{b}$ for some argument $\mathbf{b} \in \mathcal{E}$.*

   (b) *For every argument $\mathbf{b} \in \mathcal{E}$, if there exists no argument $\mathbf{a} \in \mathcal{E}'$ such that $\mathbf{a} \sqsubseteq \mathbf{b}$, then there exists no argument $\mathbf{c} \in \Delta$ such that $\mathbf{c} \sqsubseteq \mathbf{b}$.*

   Now consider each sub-case in turn:

   (a) *Every argument $\mathbf{a} \in \mathcal{E}'$ is in $\mathcal{P}$. If argument $\mathbf{a} \in \mathcal{P}$ and $\mathbf{a} \notin \Delta$ given theory context $\mathcal{C}$, then:*

   - *There must exist an elaboration $\mathbf{b} \in \mathcal{E}_A$, where $\mathcal{E}_A$ is the accepted extension of $\mathcal{C}$, such that $\mathbf{a} \sqsubset \mathbf{b}$, because procedure* reconcile *ensures that (non-strict) elaborations of arguments in a portrayal are always added to an agent's theory context, and if $\mathbf{a} \in \mathcal{P}$, but $\mathbf{a} \notin \Delta$, then $\mathbf{b} \in \mathcal{E}_A$ by the second criterion of Definition 5.2 (otherwise if $\mathbf{a} \in \mathcal{P}$ then $\mathbf{a} \in \Delta$ always).*
   - *It must be the case that $\mathbf{c} \rightharpoonup \mathbf{b}$ for some argument $\mathbf{c} \in \mathcal{P}$, but $\mathbf{c} \not\rightharpoonup \mathbf{a}$, unless there also exists an elaboration $\mathbf{d} \in \mathcal{E}_a$ such that $\mathbf{a} \sqsubset \mathbf{d} \not\sqsubseteq \mathbf{b}$ and $\mathbf{c} \not\rightharpoonup \mathbf{d}$ (from the second criterion of Definition 5.2; the first and third criteria are already fulfilled if $\mathbf{a} \in \mathcal{P}$ and $\mathbf{a}$ has not been dismissed already as invalid).*
   - *$\mathbf{b} \in \mathcal{U}$, where $\mathcal{U}$ is the unrejected argument set of $\mathcal{C}$ (because $\mathcal{E}_A \subseteq \mathcal{U}$ by Definition 3.16), so function* argue *will invoke an* attack *operation* $\mathsf{attack}(\mathcal{P}[\sigma], \mathbf{c} \rightharpoonup \mathbf{b})$*; this will replace $\mathbf{a}$ with $\mathbf{b}$ in $\mathcal{P}$, and $\mathbf{b} \in \Delta$.*

   Therefore after an invocation of argue*, for every (revised) argument $\mathbf{a} \in \mathcal{E}'$, it is the case that $\mathbf{a} \in \Delta$. Assume then that there exists no admissible extension $\mathcal{E}$ of $\mathcal{C}$ such that for each argument $\mathbf{a} \in \mathcal{E}'$, it is the case that $\mathbf{a} \sqsubseteq \mathbf{b}$ for some argument $\mathbf{b} \in \mathcal{E}$. This indicates that either:*

   i. *There is no argument $\mathbf{b} \in \mathcal{C}$ such that $\mathbf{a} \sqsubseteq \mathbf{b}$ for some argument $\mathbf{a} \in \mathcal{E}'$.*

ii. There exists no set of arguments $S \subseteq \mathcal{A}$ *of* $\mathcal{C}$ *which for each argument* $\mathbf{a} \in \mathcal{E}'$ *defends an argument* $\mathbf{b} \in \mathcal{C}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$ *(recall Definition 3.7), and which is conflict-free (recall Definition 3.8).*

As already observed, when predicate reconcile *is invoked upon the insertion of a new argument* **a**, *if there is no argument* $\mathbf{b} \in \mathcal{C}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$, *then* reconcile *adds an elaboration of* **a** *to* $\mathcal{C}$ *— thus the portrayal mechanism will rectify (i). As for (ii), we observe that if every set of arguments S which defends an elaboration* **b** *of every argument* $\mathbf{a} \in \mathcal{E}'$ *is not conflict-free, then:*

- *For every set S, there must exist an attack* $\mathbf{c} \rightharpoonup \mathbf{b}$ *for some argument* $\mathbf{c} \in S$.
- *If* $\mathbf{c} \in \mathcal{U}$, *where* $\mathcal{U}$ *is the set of unrejected arguments in* $\mathcal{C}$, *then an invocation of predicate* argue *will insert an attack* $\mathbf{d} \rightharpoonup \mathbf{e}$, *where* $\mathbf{d} \sqsubseteq \mathbf{c}$ *and* $\mathbf{a} \sqsubseteq \mathbf{e} \sqsubseteq \mathbf{b}$, *at which point* **a** *will no longer be within* $\mathcal{P}$ *and* $\mathcal{E}'$ *will no longer be a valid extension of* $\mathcal{P}$ *(because it will include both* **d** *and* **e**, *and will no longer be conflict-free).*
- *If* $\mathbf{c} \in \mathcal{R}$, *where* $\mathcal{R}$ *is the set of rejected arguments in* $\mathcal{C}$, *then either:*
  - **c** *can be dismissed given theory core* Θ *of* $\mathcal{C}$ *as per Definition 3.15, in which case* $\mathbf{a} \in \mathcal{P}$ *can be dismissed (since upon invocation of* insert *for a new argument* **f**, observe *is invoked if* **f** *is dismissable by* Θ, *elaborating* **f** *if necessary).*
  - *There must exist an argument* $\mathbf{g} \in \mathcal{E}_A$, *where* $\mathcal{E}_A$ *is the accepted extension of* $\mathcal{C}$, *such that* $\mathbf{g} \rightharpoonup \mathbf{c}$ *and an invocation of predicate* argue *will thus invoke an* attack *operation* $\mathtt{attack}(\mathbf{h} \rightharpoonup \mathbf{i})$, *where* $\mathbf{h} \sqsubseteq \mathbf{g}$ *and* $\mathbf{a} \sqsubseteq \mathbf{i} \sqsubseteq \mathbf{c}$.

  In either case, **a** *will no longer be within* $\mathcal{P}$ *and* $\mathcal{E}'$ *will no longer be a valid extension of* $\mathcal{P}$.

Therefore if there is no admissible extension $\mathcal{E}$ *of* $\mathcal{C}$ *such that for each argument* $\mathbf{a} \in \mathcal{E}'$, *it is the case that* $\mathbf{a} \in \Delta$ *of* $\mathcal{P}$ *and* $\mathbf{a} \sqsubseteq \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}$, *then either an admissible extension* $\mathcal{E}$ *will be made in* $\mathcal{C}$, *or* $\mathcal{E}'$ *will be rendered inadmissible, upon which point* reconcile *will be re-invoked until stability is reached.*

(b) Assume that there exists no admissible extension $\mathcal{E}$ *of* $\mathcal{C}$ *such that for every argument* $\mathbf{b} \in \mathcal{E}$, *if there exists no argument* $\mathbf{a} \in \mathcal{E}'$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$, *then there exists no argument* $\mathbf{c} \in \Delta$ *such that* $\mathbf{c} \sqsubseteq \mathbf{b}$. *This is easily rectified by*

inserting argument $\mathbf{c}$ *into portrayal* $\mathcal{P}$. *If the* argue *predicate is invoked, then such an argument* $\mathbf{c}$ *shall be inserted into* $\mathcal{P}$.

Therefore, if there exists a complete extension $\mathcal{E}'$ *of* $\mathcal{P}[\sigma]$ *given theory core* Θ *of* $\mathcal{C}$ *which is not a potential restriction into* Δ *of an admissible extension* $\mathcal{E}$ *of* $\mathcal{C}$, *then either* $\mathcal{E}'$ *will be made inadmissible, or arguments will be added to* $\mathcal{C}$ *such that there exists an admissible extension* $\mathcal{E}$ *as described.*

2. *Now consider the second case. Again by Definition 4.2, we can infer that there is no potential restriction* $\mathcal{E}'$ *into* $\mathcal{P}$ *where:*

   (a) *For each argument* $\mathbf{a} \in \mathcal{E}'$, *it is the case that* $\mathbf{a} \in \Delta$ *of* $\mathcal{P}$ *and* $\mathbf{a} \sqsubseteq \mathbf{b}$ *for some argument* $\mathbf{b} \in \mathcal{E}$.

   (b) *For every argument* $\mathbf{b} \in \mathcal{E}$, *if there exists no argument* $\mathbf{a} \in \mathcal{E}'$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$, *then there exists no argument* $\mathbf{c} \in \Delta$ *such that* $\mathbf{c} \sqsubseteq \mathbf{b}$.

   Again, consider the two sub-cases:

   (a) *If* $\mathbf{a} \notin \Delta$ *of* $\mathcal{P}$, *then* $\mathcal{P}$ *will be revised, as shown in the first part of this proof. Moreover, we know that any argument* $\mathbf{b} \in \mathcal{E}$ *is also an argument* $\mathbf{b} \in \mathcal{U}$, *and thus if there is no argument* $\mathbf{a} \in \mathcal{P}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$, *then we know that* argue *will insert such an argument* $\mathbf{a}$ *into* $\mathcal{P}$.

   (b) *We know that any argument* $\mathbf{c} \in \Delta$ *described as above will be added by* argue *into* $\mathcal{P}$, *ensuring that there exists an argument* $\mathbf{a} \in \mathcal{P}$ *by examination of step 1 of* argue, *such that* $\mathbf{a} \sqsubseteq \mathbf{b}$.

   Thus it is easy to see that there will be a potential restriction $\mathcal{E}'$ *of the accepted extension* $\mathcal{E}$ *of* $\mathcal{C}$. *All that remains then is to show that such a potential restriction* $\mathcal{E}'$ *is an admissible extension of* $\mathcal{P}$. *Assume otherwise. Either:*

   (a) $\mathcal{E}'$ *is not conflict-free.*

   (b) *There exists an argument* $\mathbf{a} \in \mathcal{E}'$ *such that* $\mathbf{a}$ *is not defended by* $\mathcal{E}'$.

   If $\mathcal{E}'$ *is not conflict-free, then there exists two arguments* $\mathbf{a}, \mathbf{b} \in \mathcal{E}'$ *such that* $\mathbf{a} \rightharpoonup \mathbf{b}$. *By Theorem 4.1 however, if* $\mathbf{a} \rightharpoonup \mathbf{b}$, *then there exist two arguments* $\mathbf{c}, \mathbf{d} \in \mathcal{E}$ *such that* $\mathbf{a} \sqsubseteq \mathbf{c}$, $\mathbf{b} \sqsubseteq \mathbf{d}$ *and* $\mathbf{c} \rightharpoonup \mathbf{d}$. *In that case,* $\mathcal{E}$ *is not valid accepted extension, and should be changed. Likewise, if* $\mathbf{a}$ *is not defended by* $\mathcal{E}'$, *then there exists an argument* $\mathbf{e} \in \mathcal{P}$ *such that* $\mathbf{e} \rightharpoonup \mathbf{a}$:

- *If there exists an argument* $\mathbf{f} \in \mathcal{C}$ *such that* $\mathbf{e} \sqsubseteq \mathbf{f}$*, then there must exist an argument* $\mathbf{g}$ *such that* $\mathbf{g} \rightharpoonup \mathbf{f}$ *(otherwise* $\mathcal{E}$ *does not defend against* $\mathbf{f}$*, and is thus not a valid accepted extension of* $\mathcal{C}$*). Because* Δ *of* $\mathcal{P}$ *is sufficiently expressive (Theorem 5.1),* argue *will insert an argument* $\mathbf{h}$ *into* $\mathcal{P}$ *such that* $\mathbf{h} \rightharpoonup \mathbf{e}$ *and* $\mathbf{h} \sqsubseteq \mathbf{g}$*, removing the threat to* $\mathbf{a}$ *and* $\mathcal{E}'$*.*
- *If there exists no argument* $\mathbf{f} \in \mathcal{C}$ *such that* $\mathbf{e} \sqsubseteq \mathbf{f}$*, then* reconcile *will add* $\mathbf{e}$ *to* $\mathcal{C}$*, such that accepted extension* $\mathcal{E}$ *will have to be revised unless there exists an argument* $\mathbf{g}$ *as described just above, which will be added in potential form by* argue *to* $\mathcal{P}$*.*

In any case, either an admissible extension $\mathcal{E}'$ *will be created in* $\mathcal{P}$*, or accepted extension* $\mathcal{E}$ *in* $\mathcal{C}$ *will be rendered inadmissible.*

Therefore, if the portrayal instance $\mathcal{P}[\sigma]$ *of a portrayal* $\mathcal{P}$ *with argument space* Δ *held by an agent* σ *is* not *reconciled with the theory context* $\mathcal{C}$ *of* σ*, then either* $\mathcal{P}[\sigma]$ *or* $\mathcal{C}$ *or both will be modified until reconciliation holds.*

It can be seen that the portrayal mechanism invokes reconcile whenever new information is added to the portrayal, and that argue is either invoked by reconcile, invoked by extend_arguments (upon the extension of the portrayal space as described in §5.3.2) or invoked upon changes to the theory context in general, ensuring that moves towards reconciliation are always made when appropriate.

The consequence of this is a mechanism by which the beliefs of agents engaged in interaction will inevitably move into synchrony in those areas related to the tasks at hand. Irrelevant details are left alone until such time perhaps as they become important for determining some constraint on some future interaction; attention is focused entirely on inferences which might actually affect the outcome of portrayed interactions at the present time. This is done through the lens of argumentation theory, where we view the beliefs of an agent as a persistent argumentation process conducted by the agent privately; the role of the interaction portrayal is to provide a medium through which such a private process can be efficiently influenced by the observations and assertions of peers, which of course have their own private argument systems.

In summary, this chapter has specified in full the mechanism by which interaction portrayals are constructed and maintained over the course of a protocol-driven interaction of the sort described in Chapter 2. It has demonstrated how to construct a system of arguments incrementally during dialogue, and has demonstrated how to deal with

such complications as the addition of new peers to an interaction, the observation of new empirical evidence and the introduction of invalid arguments and attacks as a consequence of mismatches in the argumentation frameworks used by individual agents to conduct argumentation. Furthermore, it has been shown that the resulting process model ensures that reconciliation of agent beliefs will eventually happen within the argument space of the resulting interaction portrayal.

However to really understand how dialogue might unfold within a group of agents involved in a portrayed interaction, we need a more in-depth demonstration of the portrayal mechanism in action.

Chapter 6

A Demonstration of a Portrayed Interaction

It will be recalled that our original motivation for this thesis' contribution lies in a desire to augment protocol-driven interactions. Specifically, we looked for a means by which to permit the spontaneous generation of dialogue between agents such that constraints on interaction could be discussed prior to their resolution. It was felt by this means we could help ensure that agents made decisions in the most enlightened manner possible.

From the perspective of the designer of a functional agent system, such a mechanism would offer two advantages. The first is that the protocols for interaction could be kept as lightweight and generic as possible — the intuition being that there would be no need to specify specific routines for discussing the properties of a given entity if there was a generic means to generate discussion automatically. The second benefit would be that the agents involved in interaction would be given an additional means to disseminate information and test their beliefs against those of peers — agents would naturally become better informed by the act of interaction, and groups of agents would develop greater common ground.

This chapter concerns itself with demonstrating the portrayal mechanism in action. In doing so, it is hoped that it is made more clear to the reader how portraying interaction can prove useful in practice. We detail what might now be considered to be the archetypical portrayed interaction, in which one agent acquires the patronage of another in order to acquire some resource from a third agent. We also consider how the interaction might unfold differently under different circumstances, and consider how such an interaction might unfold *without* the benefit of a portrayal.

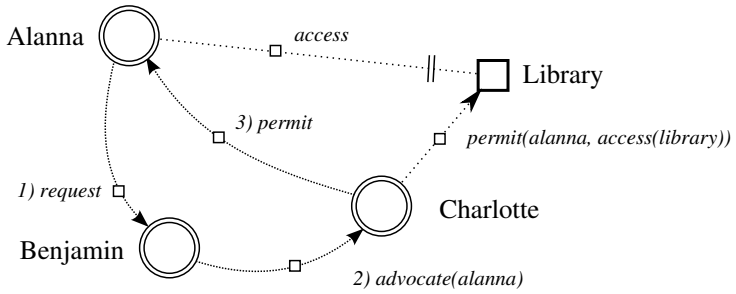


Figure 6.1: *Alanna acquires Benjamin's assistance in acquiring access to a library from Charlotte.*

6.1 An Archetypical Portrayal of Interaction

In Chapter 2 we described a simple interaction between three agents. In this interaction the first agent (named Alanna) is able to acquire access to a privileged library by petitioning the second agent (Benjamin) to recommend her to the third agent (Charlotte), that third agent having some measure of control over the library in question. In Chapters 3 and 5, we have referred to this interaction where it suited us to illustrate elements of argumentation or interaction portrayal respectively. Now we revisit this interaction in full, specifying in detail how the portrayal mechanism would augment it so as to ensure the 'best' outcome under various conditions.

6.1.1 Premise and Protocol

Within a distributed system, we have an agent named Alanna. Alanna is assigned the task of compiling a report on some topic of concern, but has observed that in order to fulfil her objective, she needs some body of additional information (which we will simply refer to as data). This can be expressed easily enough by a logical proposition:

$$\mathsf{needs}(\mathsf{alanna}, \mathsf{data})$$

We shall assume that Alanna is able, by virtue of private inference (or possibly interrogation of peers), to infer that there exists only one source of data, and that source is the eponymous library library.[1] Unfortunately, library is only accessible by a privileged few, and Alanna is not one of them. Any attempt by Alanna to access data in library will result in a refusal to reveal the requested information. Thus Alanna can easily observe that:

[1] This system of inspired nomenclature will be applied throughout this chapter.

$\neg$accessible(alanna, library)

Since Alanna is unable to simply change this fact under her own power, Alanna must enlist the assistance of peers who can.

As in any sufficiently complex society, there exist a number of established protocols by which certain classes of interaction should be conducted. The established protocol in this system for acquiring access to a restricted resource is the acquire_access protocol described originally in Chapter 1. The LCC [Robertson, 2004] protocol acquire_access allows an agent to acquire access to a privileged resource by having an existing patron of that resource recommend it to one of the resource's controllers. It defines three roles:

Applicant — An agent acting in role applicant requests the advocacy of an existing patron of a inaccessible resource. If the advocate accepts the applicant's request and the resource's controller accepts the advocate's recommendation, then success is confirmed upon being able to access the resource:

```
a(applicant(Resource), Applicant) ::
    request ⇒ a(advocate(Resource), Advocate)
    ← ¬ accessible(Applicant, Resource) ∧ patron(Advocate, Resource) then
    (   fail ← decline ⇐ a(advocate(Resource), Advocate)
        else
        fail ← reject ⇐ a(controller(Resource), Controller)
        else
        succeed ← permit ⇐ a(controller(Resource), Controller) ∧
                        accessible(Applicant, Resource)   ).
```

Advocate — The agent designated the advocate will present an applicant's case to the resource's controller should it consider the applicant trustworthy:

```
a(advocate(Resource), Advocate) ::
    request ⇐ a(applicant(Resource), Applicant) then
    (   advocate(Applicant) ⇒ a(controller(Resource), Controller)
        ← controller(Controller, Resource) ∧ trustworthy(Applicant, access(Resource))
        else
        decline ⇒ a(applicant(Resource), Applicant)   ).
```

Controller — The controller permits access to a resource only if an applicant is both eligible and has been advocated by a peer it already trusts:

a(controller(*Resource*), *Controller*) ::
advocate(*Applicant*) $\Leftarrow$ a(advocate(*Resource*), *Advocate*) then
(permit(*Applicant*, access(*Resource*))
$\leftarrow$ controls(*Controller*, permissions(*Resource*)) $\wedge$
trusts(*Controller*, *Advocate*) $\wedge$ eligible(*Applicant*, access(*Resource*)) then
permit $\Rightarrow$ a(applicant(*Resource*), *Applicant*))
else
reject $\Rightarrow$ a(applicant(*Resource*), *Applicant*)).

As to *why* this is the established protocol for such activity, there could be any number of reasons. Perhaps there has been a history of agents abusing resources, and it has been agreed by the controllers of those resources that new agents will only be permitted access if they have the support of an existing user. Perhaps the above protocol evolved naturally from the tendency of applicants to ask for assistance from friendly peers, who found the best way to assist if they already had access themselves was to simply ask more powerful peers to grant the applicant the requested privileges. Perhaps it was simply forced upon the system from on high. In any case, this protocol makes a good exemplar for demonstrating portrayals for two reasons:

- The protocol is simple, but not *too* simple. In particular, the interaction requires the coordination of three peers — two peers simply would not demonstrate well how a portrayal serves as an intermediary between multiple agent theories, nor would it sufficiently demonstrate the support for asynchronicity of the portrayal mechanism specified in Chapter 5.

- The protocol is dependent on a number of high level, variably subjective constraints which can be resolved in many different ways in many different domains. These constraints are therefore good examples of the kind of constraint for which additional discussion between peers can have a significant benefit.

In any case, this interaction protocol is capable of granting Alanna her desire. The consequent interaction can then be seen to have four basic parts:

1. Initiation of the interaction by Alanna, which includes establishing the inaccessibility of the library and the selection of a patron (Benjamin) to act as her advocate.

2. Benjamin taking on the role of Alanna's advocate, which requires that Benjamin

knows the library controller (Charlotte) and that he is satisfied that Alanna can be trusted with access to the library.

3. Charlotte accepting Benjamin's recommendation, which requires that Charlotte indeed has control over the library, that Benjamin is trusted in his given role and that Alanna is eligible to be granted access.

4. Confirmation of access granted, being dependent on a response by Charlotte and an indication that the environment has changed in Alanna's favour.

Without further delay, let us consider the start of interaction.

6.1.2 Initiating Interaction

The first thing Alanna needs to do is to initiate interaction (as per §2.3.1), selecting protocol acquire_access, associating herself with the role of applicant and associating the resource in question with library:

a(applicant(library), alanna) ::
request $\Rightarrow$ a(advocate(library), *Advocate*)
$\leftarrow \neg$ accessible(alanna, library) $\wedge$ patron(*Advocate*, library) then ...

Alanna then needs only to satisfy the first pair of constraints before dispatching a message to her chosen advocate. Let us briefly consider each proposition in turn:

accessible(*Agent*, *Resource*) — accessible is an example of an objective predicate which can typically be determined by direct observation on the part of *Agent* — as a result, we would expect that argumentation would be unnecessary here, especially in this case where the constraint will be resolved by *Agent* itself. Nevertheless if interaction is successful, then we can expect the state of this proposition to change, and a portrayal will internalise this and portray any relevant consequences (see 6.1.4 below).

patron(*Agent*, *Resource*) — patron is an example of a domain-specific predicate which may or may not be easy to determine. Essentially, we do not know what a 'patron' of a given resource *is* in a generic sense; instead we rely on agents in a given domain knowing for themselves. Such reliance is necessary for protocols not tied to a specific problem domain.

We designate Alanna's beliefs as theory Π_A. The context $\mathcal{C}_A$ for Π_A can be described by an argumentation process conducted within an assumption-based argumentation framework $(\mathcal{L}, \vdash, \Delta)_A$, with a theory core Θ_A (as per Definition 3.16). Let us assume that from $\mathcal{C}_A$, Alanna can construct the following accepted arguments such that the set of supporting assumptions are all in Π_A:

$$\begin{aligned}
\mathbf{a}_A &= \langle\ \{\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \},\ \neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})\ \rangle \\
\mathbf{b}_A &= \langle\ \{\ \mathsf{access}(\mathsf{benjamin},\mathsf{library}),\\
&\qquad \forall X,Y.\ \mathsf{access}(X,Y) \rightarrow \mathsf{patron}(X,Y)\ \},\\
&\quad\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \rangle \\
\mathbf{c}_A &= \langle\ \{\ \mathsf{access}(\mathsf{dante},\mathsf{library}),\\
&\qquad \forall X,Y.\ \mathsf{access}(X,Y) \rightarrow \mathsf{patron}(X,Y)\ \},\\
&\quad\ \mathsf{patron}(\mathsf{dante},\mathsf{library})\ \rangle
\end{aligned}$$

We justify argument $\mathbf{a}_A$ by observation — Alanna is able to directly determine that library is inaccessible to her, and thus does not need to justify such a claim further (i.e. $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library}) \in \Theta_A$). Arguments $\mathbf{b}_A$ and $\mathbf{c}_A$ are standard arguments based on the assumptions that Benjamin and Dante can access library and that any agent which can access a resource is a patron of that resource — these arguments have proven to be admissible within $\mathcal{C}_A$. Any counter-arguments that Alanna can conceive within the argument space Δ of the argumentation framework within $\mathcal{C}_A$ have either been defeated, or Alanna has simply chosen $\mathbf{b}_A$ and $\mathbf{c}_A$ over other, equally admissible arguments — we will not concern ourselves with the precise details at this time.

At this point, in accordance with §5.2, Alanna can conceive a new portrayal focused on the two propositions $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ and $\mathsf{patron}(X,\mathsf{library})$ (all other constraints specified by acquire_access require further development of the interaction before they will be portrayed, not least the adoption of the roles they are found in). Executing $\mathsf{conceive}(\mathcal{C}_A, \{\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library}), \mathsf{patron}(X,\mathsf{library})\}, \mathcal{P}[\mathsf{alanna}])$ (as in §5.2.2) will produce a portrayal instance $\mathcal{P}[\mathsf{alanna}]$ where:

- $\Sigma = \{\mathsf{alanna}\}$
- $\Upsilon = \{\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library}), \mathsf{patron}(X,\mathsf{library})\}$
- $\mathcal{A} = \{\mathbf{a}', \mathbf{b}', \mathbf{c}'\}$ (see below); there are no attacks between arguments in $\mathcal{A}$.
- $\mathsf{inv}_{\mathcal{A}}(\mathsf{alanna}) = \mathsf{inv}_{\rightarrow}(\mathsf{alanna}) = \mathsf{obs}(\mathsf{alanna}) = \emptyset$.
- $\mathsf{acc}(\mathsf{alanna}) = \{\mathbf{a}', \mathbf{b}', \mathbf{c}'\}$.

The set of arguments $\mathcal{A}$ of $\mathcal{P}[\mathsf{alanna}]$ is the potential restriction of arguments $\mathbf{a}_A$, $\mathbf{b}_A$ and $\mathbf{c}_A$ into the argument space of $\mathcal{P}$ (as per Definition 4.2). This results in a set of trivial 'identity' arguments:

$\mathbf{a}' = \langle \{ \neg\mathsf{accessible}(\mathsf{alanna}, \mathsf{library}) \}, \neg\mathsf{accessible}(\mathsf{alanna}, \mathsf{library}) \rangle$
$\mathbf{b}' = \langle \{ \mathsf{patron}(\mathsf{benjamin}, \mathsf{library}) \}, \mathsf{patron}(\mathsf{benjamin}, \mathsf{library}) \rangle$
$\mathbf{c}' = \langle \{ \mathsf{patron}(\mathsf{dante}, \mathsf{library}) \}, \mathsf{patron}(\mathsf{dante}, \mathsf{library}) \rangle$

We keep arguments simple because they might not be disputed — we wish to refrain from argumentation which probably will not have effect on the outcome of interaction. If more detail is required later, than we can elaborate upon arguments on demand. Of course, there is always the possibility that despite being in apparent agreement about the correct resolution of a constraint, more detailed examination of arguments will reveal cause for belief revision. We can only be certain that this is not the case however by comparing the belief systems of agents in their entirety, which for real-world systems can be expected to be prohibitively expensive computationally. Pragmatically though, we do not need such assurance — as long as the consequences of portrayal are *in the worst case* equal to the case where no argumentation is performed at all, even heavily restricted argumentation is adequate for our purposes.

Portrayals permit a sceptical agent to posit undecided arguments — this allows an agent which prefers not to commit to controversial arguments to still critique the arguments of peers even where it does not necessarily accept the attack either (otherwise it will not always be possible to reconcile context with portrayal). Even at the point of portrayal conception, a sceptical or uncertain agent can posit a set of arguments which is not conflict-free with an aim to see how peers later respond. In Example 5.5, we described an alternative to the above scenario in which agent Alanna posited an additional argument into the portrayal:

$\mathbf{d}' = \langle \{ \neg\mathsf{patron}(\mathsf{dante}, \mathsf{library}) \}, \neg\mathsf{patron}(\mathsf{dante}, \mathsf{library}) \rangle$

In this scenario, Alanna is able to infer both that Dante is and is not a patron of library, and is undecided as to which to accept. As such, she posits both arguments into the portrayal and notes that they rebut — in doing so she invites peers to attack or accept either assertion in the hope that evidence is provided sufficient to decide the matter to her own satisfaction.

Whilst normally arguments in a new portrayal will be expressed as initially trivially as possible, it is possible for initial portrayal arguments to begin in an already partially elaborated state if there exists dispute between arguments. Imagine that in C_A, argument $\mathbf{b}_A$ above was expressed instead as follows:

$$\mathbf{b}_A = \langle\ \{\ \mathsf{access}(\mathsf{dante},\mathsf{library}),\ \forall X,Y.\ \mathsf{access}(X,Y) \rightarrow \mathsf{patron}(X,Y),\ \mathsf{patron}(\mathsf{dante},\mathsf{library}) \rightarrow \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \},\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \rangle$$

If we also assume that $\mathbf{d}'$ is within $\mathcal{P}[\mathsf{alanna}]$, then $\mathbf{d}' \rightharpoonup \mathbf{b}_A$ according to $\mathcal{C}_A$, but $\mathbf{d}' \not\rightharpoonup \mathbf{b}'$. By Definition 5.2 (second criterion) then, $\mathbf{b}' \notin \Delta$, because $\mathbf{b}'$ does not acknowledge the necessary conflict with argument $\mathbf{d}'$ (however if *both* versions of $\mathbf{b}_A$ were in $\mathcal{C}_A$, then $\mathbf{b}'$ *would* be in Δ, because Alanna can claim that Benjamin is a patron of the library without necessarily rejecting $\mathbf{d}_A$ first). In this case, we would instead posit argument $\mathbf{b}''$ into $\mathcal{P}[\mathsf{alanna}]$:

$$\mathbf{b}'' = \langle\ \{\ \mathsf{patron}(\mathsf{dante},\mathsf{library}),\ \mathsf{patron}(\mathsf{dante},\mathsf{library}) \rightarrow \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \},\ \mathsf{patron}(\mathsf{benjamin},\mathsf{library})\ \rangle$$

This would then give us a portrayal containing arguments $\mathbf{a}'$, $\mathbf{b}''$, $\mathbf{c}'$ and $\mathbf{d}'$, where $\mathbf{c}' \rightharpoonup \mathbf{d}'$, $\mathbf{d}' \rightharpoonup \mathbf{c}'$ and $\mathbf{d}' \rightharpoonup \mathbf{b}''$.

For the remainder of this example however, we shall assume that the original portrayal instance $\mathcal{P}[\mathsf{alanna}]$ specified earlier (without argument $\mathbf{d}_A$ or the alternative $\mathbf{b}_A$) is the one used. In this case, Alanna can easily determine that $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ and $\mathsf{patron}(\mathsf{benjamin},\mathsf{library})$, and will thus dispatch a message request to Benjamin:

```
a(applicant(library), alanna) ::
    c(request ⇒ a(advocate(library), benjamin)) then ...
```

This brings Benjamin into the interaction.

6.1.3 Establishing Advocacy

Let us assume then that Benjamin is inducted into the interaction in the role of advocate. We can now produce a model for Benjamin's contribution to dialogue:

```
a(advocate(library), benjamin) ::
    request ⇐ a(applicant(library), alanna) then
    (   advocate(alanna) ⇒ a(controller(library), Controller)
        ← controller(Controller, library) ∧ trustworthy(alanna, access(library))
        else
        decline ⇒ a(applicant(library), alanna)   ).
```

It befalls Benjamin to determine the controller of library and the trustworthiness of Alanna. Again, let us consider the available propositions:

controller(*Agent*, *Resource*) — controller is another objective predicate which can in most domains be determined by direct observation. The most likely complication is simply that a given agent might not be able to identify *Agent* for itself and may have to ask other peers.

trustworthy(*Agent*, *Action*) — trustworthy is a highly subjective predicate, the resolution of which is primarily subject to the preferences of the resolving agent. This makes it an excellent candidate for portrayal — agents can state their basis for considering *Agent* to be trustworthy (or not) in the context of *Action* and can point out perceived flaws in each other's reasoning. Because the predicate is so subjective however, it is quite possible the end result of argumentation will be a set of rebutting arguments all equally admissible — the final decision will then lie with the agent assigned the proposition to satisfy. Nevertheless, there is still value in laying out the different arguments and their internal assumptions, if only for the information disseminated in doing so.

Before Benjamin can begin to examine the options available to him however, he must establish his own portrayal instance $\mathcal{P}[\mathsf{benjamin}]$. In accordance with §5.3.1, Alanna will send a copy of $\mathcal{P}[\mathsf{alanna}]$ to Benjamin alongside message request; Benjamin will then invoke validate$(\mathcal{P}[\mathsf{benjamin}])$ (where $\mathcal{P}[\mathsf{benjamin}] = \mathcal{P}[\mathsf{alanna}]$). For each argument in the portrayal, Benjamin will check three things:

1. Does the argument conflict with Benjamin's observations?
2. Is the argument logically invalid?
3. Are the attacks associated with the argument logically invalid?

Since the portrayal is that this point very simple, the answer to all three questions is no for all arguments. Benjamin can proceed then to invoke reconcile$(\mathcal{P}[\mathsf{benjamin}], \mathcal{A})$, where $\mathcal{A}$ is the set of arguments in $\mathcal{P}$.

The role of the reconcile procedure (§5.4.11) is to check the arguments in a portrayal with the private arguments an agent can conceive within its own theory context, and to trigger further argumentation. The first thing to do is to extract the arguments in $\mathcal{P}$ which are *not* represented in Benjamin's beliefs — in this case, only argument $\mathbf{a}'$. Having found such an argument, Benjamin's theory context $\mathcal{C}_B$ is then revised accordingly:

- R is the set of all arguments $\mathbf{b} \in \mathcal{C}_B$ such that $\mathbf{b} \sqsubset \mathbf{a}'$ — in this case $R = \emptyset$.

- The argument space Δ of $\mathcal{C}_B$ is extended to include $\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library})$ as per Definition 4.4 — in this case, this merely entails adding the proposition to the set of hypotheses from which arguments can be constructed.

- $\mathcal{A}$ in $\mathcal{C}_B$ is replaced with the potential expansion $\mathcal{A} \cup \{\mathbf{a}'\}$ into (the extended) Δ as per Definition 4.3 — in this case, this merely entails adding $\mathbf{a}'$ to $\mathcal{A}$, and factoring in any conflicts (of which there are none).

Benjamin must then establish which arguments he currently accepts. It can be seen that $\mathsf{acc}(\mathsf{benjamin}) = \{\mathbf{a}',\mathbf{b}',\mathbf{c}'\}$, just as for Alanna; thus the reconcile procedure will execute $\mathsf{accept}(\mathcal{P}[\mathsf{benjamin}],\{\mathbf{a}',\mathbf{b}',\mathbf{c}'\})$ (specified in §5.4.10) which will update the acceptance function acc in all portrayal instances. Operation accept will be invoked whenever an agent finds itself forced to change its beliefs during reconciliation.

Benjamin must now invoke $\mathsf{argue}(\mathcal{P}[\mathsf{benjamin}])$ (§5.4.1). In this scenario, Benjamin accepts all of Alanna's arguments so far, but is able to posit an additional argument:

$\mathbf{d}' = \langle\ \{\ \mathsf{patron}(\mathsf{eliza},\mathsf{library})\ \},\ \mathsf{patron}(\mathsf{eliza},\mathsf{library})\ \rangle$

Argument $\mathbf{d}'$ seems rather extraneous. After all, the constraint to which it is attached has already been resolved. This illustrates a difficulty in the design of the portrayal mechanism — at which point should an agent stop making suggestions? After all, if this interaction fails, Alanna might wish to try again with a different patron, at which point knowledge of additional patrons might be very useful. It may also be possible in some interaction protocols to backtrack, and try different solutions as they emerge.[2] In many practical circumstances however, there is probably a limit to how many different instances of a proposition should be suggested within a portrayal — but the exact determination of this limit is really a matter for the individual implementation, since considered in the abstract, any limit would be arbitrary. For this thesis, we err on the side of permissiveness.

Now that there is more than one agent in the interaction, any changes to the portrayal must be communicated to all peers. The posit operation (§5.4.2) is used to dispatch a message $\mathsf{posit}(\mathbf{d}')$ to Alanna, who then invokes insert to insert the argument into $\mathcal{P}[\mathsf{alanna}]$. insert acts just as validate, but for single arguments; thus it also enacts reconcile, then argue. In this case, Alanna has nothing further to add, and merely responds by accepting $\mathbf{d}'$ into $\mathsf{acc}(\mathsf{alanna})$.

[2] LCC by default does *not* allow this, but can be extended to do so [Osman, 2003].

Having acquired and updated $\mathcal{P}[\mathsf{benjamin}]$, Benjamin can now attend to his own role in interaction. Having adopted the advocate role, Benjamin must re-portray his role so as to identify any new portrayable propositions (as per §5.3) — Benjamin will need to invoke extend_space($\mathcal{P}[\mathsf{benjamin}], \{\mathsf{controller}(X,\mathsf{library}), \mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\}$).

extend_space works by sending a message to all peers, permitting them to immediately start making arguments regarding the given propositions. In this case, Alanna can make the following argument within $\mathcal{C}_A$:

$$\begin{aligned}\mathbf{e}_A = \langle\ \{\ & \mathsf{tenure}(\mathsf{alanna},\mathsf{edinburgh}),\\ & \mathsf{university}(\mathsf{edinburgh}),\\ & \forall X,Y.\ \mathsf{tenure}(X,Y) \wedge \mathsf{university}(Y) \rightarrow \mathsf{researcher}(X),\\ & \mathsf{needs}(\mathsf{alanna},\mathsf{data}),\\ & \mathsf{only\_source}(\mathsf{library},\mathsf{data}),\\ & \forall X,Y,Z.\ \mathsf{needs}(X,Y) \wedge \mathsf{only\_source}(Z,Y) \rightarrow \neg\mathsf{beneficial}(X,\mathsf{abuse}(\mathsf{access}(Z))),\\ & \forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X,Y)\ \},\\ & \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle\end{aligned}$$

. . . and so Alanna can posit argument $\mathbf{e}'$ in $\mathcal{P}$:

$$\mathbf{e}' = \langle\ \{\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},\ \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$$

Meanwhile, Benjamin can make the following arguments in $\mathcal{C}_B$:

$$\mathbf{f}_B = \langle\ \{\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \},\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \rangle$$

$$\begin{aligned}\mathbf{g}_B = \langle\ \{\ & \mathsf{abused}(\mathsf{alanna},\mathsf{access}(\mathsf{laboratory})),\\ & \mathsf{analogous}(\mathsf{access}(\mathsf{laboratory}),\mathsf{access}(\mathsf{library})),\\ & \forall X,Y,Z.\ \mathsf{abused}(X,Y) \wedge \mathsf{analogous}(Y,Z) \rightarrow \neg\mathsf{trustworthy}(X,Z)\ \},\\ & \neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle\end{aligned}$$

Thus Benjamin asserts the following into $\mathcal{P}$:

$$\mathbf{f}' = \langle\ \{\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \},\ \mathsf{controller}(\mathsf{charlotte},\mathsf{library})\ \rangle$$

$$\mathbf{g}' = \langle\ \{\ \neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \},\ \neg\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle$$

Both peers now have to reconcile the other's argument(s). Upon reception of $\mathbf{e}'$, Benjamin determines that $\mathbf{e}'$ is potentially another argument in $\mathcal{C}_B$:

$$\begin{aligned}\mathbf{e}_B = \langle\ \{\ & \mathsf{postdoc}(\mathsf{alanna}),\\ & \forall X.\ \mathsf{postdoc}(X) \rightarrow \mathsf{researcher}(X),\\ & \neg\mathsf{beneficial}(\mathsf{alanna},\mathsf{abuse}(\mathsf{access}(\mathsf{library}))),\\ & \forall X,Y.\ \mathsf{researcher}(X) \wedge \neg\mathsf{beneficial}(X,\mathsf{abuse}(Y)) \rightarrow \mathsf{trustworthy}(X,Y)\ \},\\ & \mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\ \rangle\end{aligned}$$

This argument is attacked within $\mathcal{C}_B$ however by yet another argument:

$$\begin{aligned}\mathbf{h}_B = \langle\ \{\ & \neg\exists X.\ \mathsf{research\_topic}(\mathsf{alanna},X),\\ & \forall X.\ \mathsf{researcher}(X) \rightarrow \exists Y.\ \mathsf{research\_topic}(X,Y)\ \},\\ & \neg\mathsf{researcher}(\mathsf{alanna})\ \rangle\end{aligned}$$

Thus, when Benjamin invokes argue($\mathcal{P}$[benjamin]), he will as a consequence invoke attack($\mathcal{P}$[benjamin], $\mathbf{h}' \rightharpoonup \mathbf{e}''$) (§5.4.4), introducing the following arguments into $\mathcal{P}$:

$\mathbf{e}'' = \langle$ { researcher(alanna),
¬beneficial(alanna, abuse(access(library))),
$\forall X, Y.$ researcher(X) ∧ ¬beneficial(X, abuse(Y)) → trustworthy(X, Y) },
trustworthy(alanna, access(library)) $\rangle$
$\mathbf{h}' = \langle$ { ¬researcher(alanna) }, ¬researcher(alanna) $\rangle$

When Alanna receives message attack($\mathbf{h}' \rightharpoonup \mathbf{e}''$), she executes insert($\mathcal{P}$[alanna], $\mathbf{e}''$), insert($\mathcal{P}$[alanna], $\mathbf{h}'$) and insert_attack($\mathcal{P}$[alanna], $\mathbf{h}' \rightharpoonup \mathbf{e}''$) in turn:

- Upon invoking insert($\mathcal{P}$[alanna], $\mathbf{e}''$), argument $\mathbf{e}'$ in $\mathcal{P}$[alanna] will be replaced by $\mathbf{e}''$, since $\mathbf{e}' \sqsubset \mathbf{e}''$. Because $\mathbf{e}''$ is still potentially argument $\mathbf{e}_A$ within $\mathcal{C}_A$, reconciliation is trivial; because $\mathbf{e}_A$ is accepted within $\mathcal{C}_A$, no counter-argument is made.

- Upon invoking insert($\mathcal{P}$[alanna], $\mathbf{h}'$), argument $\mathbf{h}'$ is inserted into $\mathcal{P}$[alanna]. Argument $\mathbf{h}'$ can be refuted by observation however (Alanna knows that she is a researcher), so Alanna invokes observe($\mathcal{P}$[alanna], researcher(alanna)) (§5.4.6).

- Upon invoking insert_attack($\mathcal{P}$[alanna], $\mathbf{h}' \rightharpoonup \mathbf{e}''$), Alanna ensures that $\mathbf{h}' \rightharpoonup \mathbf{e}''$ is established in $\mathcal{P}$[alanna]. Alanna also checks that $\mathbf{h}' \rightharpoonup \mathbf{e}''$ is a valid attack according to $\mathcal{C}_A$ (it is, even if Alanna rejects $\mathbf{h}'$).

The effect of observe($\mathcal{P}$[alanna], researcher(alanna)) is to send a message to Benjamin to the effect that researcher(alanna) $\in \Theta_A$. Benjamin can then choose to either treat researcher(alanna) as part of his own theory core Θ_B, or treat it as an influence on his own preferences when interpreting the arguments within $\mathcal{C}_B$. In this case, Benjamin takes the viewpoint that Alanna is a sincere, competent agent, and shall add researcher(alanna) to Θ_B.

It might be wondered why an attacker *presumes* the elaboration of a given defending argument rather than asking for an elaboration from the original positing agent. Basically, this approach reduces the amount of dialogue between peers (no need to ask for elaboration before attack), and solves the problem of determining *how far* to elaborate upon a given argument. Imagine an argument which, within an agent's theory context, involves a substantial chain of inference from base premises to the final claim. Given a potential argument and a request for elaboration, how does an agent determine the extent to which it lays out its reasoning? For substantial examples, a full elaboration may be cumbersome and invite a lot of dispute that perhaps will have no

effect on the final result. There may also be considerable variation between how agents infer their beliefs from base premises despite general agreement as to end conclusions; in an open system, the existence of a shared protocol suggests a common ontology for explicit constraints, but that shared ontology may decay the further agents examine each other's reasoning. Using incremental elaboration may simply lead to an agent being requested multiple times to elaborate upon an argument. Since for synchronisation an agent need only elaborate upon claims which another agent is certain can only be made using certain assumptions which it can itself attack, we can choose an aggressive form of elaboration whereupon an attacker presumes and a defender either concedes, counters or provides an alternative explanation; this will exhibit incomplete behaviour as far as finding every way that one agent's beliefs can attack another's, but would still provide ample opportunity for improving the outcome of interaction in many situations.

Concurrent to Benjamin receiving argument $\mathbf{e}'$, Alanna receives arguments $\mathbf{f}'$ and $\mathbf{g}'$. Alanna accepts $\mathbf{f}'$ without question, but disputes $\mathbf{g}'$. Within $\mathcal{C}_A$, argument $\mathbf{g}'$ is potentially another argument:

$$\begin{aligned}\mathbf{g}_A = \langle\ \{\ & \exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X)),\\ & \forall X, Y.\ \mathsf{trustworthy}(\mathsf{X}, \mathsf{access}(\mathsf{Y})) \rightarrow \neg\exists Z.\ \mathsf{abused}(X, \mathsf{access}(Z))\ \},\\ & \forall X.\ \neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{X}))\ \rangle\end{aligned}$$

Alanna can defeat this argument with another:

$$\begin{aligned}\mathbf{i}_A = \langle\ \{\ & \forall X.\ \neg\mathsf{evidence}(\mathsf{alanna}, \mathsf{abuse}(\mathsf{access}(X))),\\ & \forall X, Y.\ \mathsf{abused}(X, Y) \rightarrow \mathsf{evidence}(X, \mathsf{abuse}(Y))\ \},\\ & \neg\exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X))\ \rangle\end{aligned}$$

Thus, when Alanna invokes $\mathsf{argue}(\mathcal{P}[\mathsf{alanna}])$, she will introduce arguments $\mathbf{g}''$ and $\mathbf{i}'$ into $\mathcal{P}$:

$$\begin{aligned}\mathbf{g}'' = \langle\ \{\ & \exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X)),\\ & \forall X, Y.\ \mathsf{trustworthy}(\mathsf{X}, \mathsf{access}(\mathsf{Y})) \rightarrow \neg\exists Z.\ \mathsf{abused}(X, \mathsf{access}(Z))\ \},\\ & \forall X.\ \neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{X}))\ \rangle\\ \mathbf{i}' = \langle\ \{\ & \neg\exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X))\ \},\ \neg\exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X))\ \rangle\end{aligned}$$

Upon receiving message $\mathsf{attack}(\mathbf{i}' \rightharpoonup \mathbf{g}'')$:

- Benjamin will replace $\mathbf{g}'$ with $\mathbf{g}''$ in $\mathcal{P}[\mathsf{benjamin}]$. However because $\mathbf{g}'' \not\sqsubseteq \mathbf{g}_B$, argument $\mathbf{g}''$ will be inserted into $\mathcal{C}_B$ as a new argument (courtesy of reconcile) and Benjamin will invoke $\mathsf{posit}(\mathcal{P}[\mathsf{benjamin}], \mathbf{j}')$ (courtesy of argue), where $\mathbf{j}'$ is a slightly more explicated re-iteration of Benjamin's original objection (see below).

- Benjamin will insert $\mathbf{i}'$ into $\mathcal{P}[\mathsf{benjamin}]$ and into $\mathcal{C}_B$; upon reconciling $\mathbf{i}'$ with $\mathcal{C}_B$, Benjamin realises that $\mathbf{i}' \rightharpoonup \mathbf{j}'$, and so invokes $\mathsf{attack}(\mathcal{P}[\mathsf{benjamin}], \mathbf{i}' \rightharpoonup \mathbf{j}')$ and revises his beliefs accordingly.

- Benjamin will establish that $\mathbf{i}' \rightharpoonup \mathbf{g}''$ in $\mathcal{P}[\mathsf{benjamin}]$.

Whilst Benjamin concedes that validity of Alanna's attack on argument $\mathbf{g}'$, Benjamin still has a rebuttal for $\mathbf{e}$. This is argument $\mathbf{j}'$:

$$\begin{aligned}\mathbf{j}' = \langle\ \{\ & \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(\mathsf{laboratory})),\\ & \mathsf{analogous}(\mathsf{access}(\mathsf{laboratory}), \mathsf{access}(\mathsf{library})),\\ & \forall X, Y, Z.\ \mathsf{abused}(X,Y) \wedge \mathsf{analogous}(Y,Z) \rightarrow \neg\mathsf{trustworthy}(X,Z)\ \},\\ & \neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\ \rangle\end{aligned}$$

Whereas $\mathbf{j}'$ would originally have been too detailed for the argument space Δ of $\mathcal{P}$, the elaboration of $\mathbf{g}'$ into an argument no longer representative of Benjamin's private beliefs means that now $\mathbf{j}' \in \Delta$. However, argument $\mathbf{j}'$ is already attacked by $\mathbf{i}'$ in $\mathcal{P}$, which Benjamin realises when he integrates $\mathbf{i}'$ into his own beliefs. Thus Benjamin is forced to follow up on his positing of $\mathbf{j}'$ with the admission that $\mathbf{i}' \rightharpoonup \mathbf{j}'$ — if he does not do this, Alanna will do it instead (in fact, Alanna might do it anyway if there is a significant gap between receiving $\mathbf{j}'$ and notification that $\mathbf{i}' \rightharpoonup \mathbf{j}'$, at which point each agent would receive the other's attack notification to no consequence).

Thus at this point we have a portrayal $\mathcal{P}$, where for both instances $\mathcal{P}[\mathsf{alanna}]$ and $\mathcal{P}[\mathsf{benjamin}]$:

- $\Sigma = \{\mathsf{alanna}, \mathsf{benjamin}\}$.

- $\Upsilon = \{\neg\mathsf{accessible}(\mathsf{alanna}, \mathsf{library}), \mathsf{patron}(X_1, \mathsf{library}), \mathsf{controller}(X_2, \mathsf{library}),$ $\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\}$.

- $\mathcal{A} = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}', \mathbf{e}'', \mathbf{f}', \mathbf{g}'', \mathbf{h}', \mathbf{i}', \mathbf{j}'\}$, where $\mathbf{e}'' \rightharpoonup \mathbf{g}''$, $\mathbf{e}'' \rightharpoonup \mathbf{j}'$, $\mathbf{g}'' \rightharpoonup \mathbf{e}''$, $\mathbf{h}' \rightharpoonup \mathbf{e}''$, $\mathbf{i}' \rightharpoonup \mathbf{g}''$, $\mathbf{i}' \rightharpoonup \mathbf{j}'$ and $\mathbf{j}' \rightharpoonup \mathbf{e}''$.

- $\mathsf{inv}_{\mathcal{A}}(\mathsf{alanna}] = \mathsf{inv}_{\mathcal{A}}[\mathsf{benjamin}) = \emptyset$.

- $\mathsf{inv}_{\rightharpoonup}(\mathsf{alanna}] = \mathsf{inv}_{\rightharpoonup}[\mathsf{benjamin}) = \emptyset$.

- $\mathsf{obs}(\mathsf{alanna}) = \{\mathsf{researcher}(\mathsf{alanna})\}$ and $\mathsf{obs}(\mathsf{benjamin}) = \emptyset$.

- $\mathsf{acc}(\mathsf{alanna}) = \mathsf{acc}(\mathsf{benjamin}) = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}', \mathbf{e}'', \mathbf{f}', \mathbf{i}'\}$.

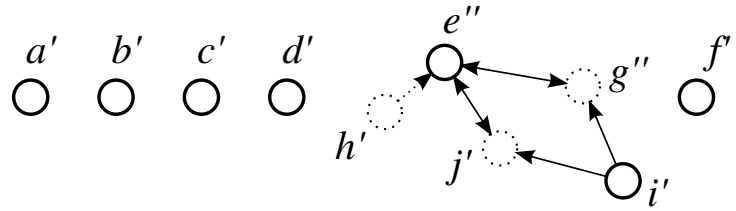


Figure 6.2: *The accepted interpretation of the system of arguments within the interaction portrayal of §6.1.3.*

Assuming that Benjamin dismisses argument $\mathbf{h}'$ on the basis that $\mathsf{researcher}(\mathsf{alanna}) \in \Theta_A$ (essentially adding $\mathsf{researcher}(\mathsf{alanna})$ to Θ_B), then both agents accept argument $\mathbf{e}'$, which means that both agents accept that $\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$. If Benjamin had not dismissed $\mathbf{h}'$, then Benjamin would have rejected $\mathbf{e}'$. Moreover, if Benjamin had been able to attack $\mathbf{i}'$, or if Benjamin had another argument by which it could claim that $\neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ that was not attacked by $\mathbf{i}'$, then Benjamin would have accepted that claim instead.

Given the portrayal $\mathcal{P}$ above however, and assuming that the declared accepted extension of $\mathcal{P}$ reflects the accepted beliefs of Benjamin, then Benjamin will unfold his role in interaction as follows:

a(advocate(library), benjamin) ::
    c(request ⇐ a(applicant(library), alanna)) then
    c(advocate(alanna) ⇒ a(controller(library), charlotte)).

If Benjamin had decided that $\neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$, or even if Benjamin found himself unable to decide either $\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ *or* $\neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ (i.e. Benjamin simply did not know which was true), then Benjamin would instead have unfolded his role in interaction as so:

a(advocate(library), benjamin) ::
    c(request ⇐ a(applicant(library), alanna)) then
    c(decline ⇒ a(applicant(library), alanna)).

At which point Alanna would receive the message decline from Benjamin, forcing her to bring interaction to a close having failed to acquire access to library. In either case, this brings an end to Benjamin's role in interaction — as specified in §5.3.4, Benjamin will indicate the completion of his role to his peers. Benjamin may still contribute to the portrayal, but once all agents have finished their roles, the interaction will end and the portrayal will be disposed of.

6.1.4 Granting Permission

Let us assume that Benjamin *did* consider Alanna to be worthy of trust. In this case, Benjamin recommends Alanna to Charlotte, the accepted controller of access to library. As controller, Charlotte has the right to accept or reject Alanna's application:

a(controller(library), charlotte) ::
advocate(alanna) $\Leftarrow$ a(advocate(library), benjamin) then
(permit(alanna, access(library))
$\leftarrow$ controls(charlotte, permissions(library)) $\wedge$
trusts(charlotte, benjamin) $\wedge$ eligible(alanna, access(library)) then
permit $\Rightarrow$ a(applicant(library), alanna))
else
reject $\Rightarrow$ a(applicant(library), alanna)).

If Charlotte is indeed able to manipulate permissions for library, and if Charlotte trusts Benjamin, then Alanna will be granted permission to access library provided that she is eligible. Once more, we look at the propositions to be satisfied:

controls(*Agent*, *Configuration*) — controls is an objective predicate used in this protocol to confirm that *Agent* has the power to fulfil its role, essentially verifying the satisfaction of the controller predicate earlier.

trusts($Agent_1$, $Agent_2$) — trusts is an abstract predicate which is sensitive to the context in which it is invoked. In particular, trusts is distinguished from the predicate trustworthy in that rather than establishing the *suitability* for trust in a given context, trusts merely confirms that trust has already been given. In this case, the role of this proposition is to confirm that the agent in the role of advocate has the influence to recommend the applicant agent in the first place. Yet again, the details are left out of the predicate and left to the discretion of individual agents.

eligible(*Agent*, *Action*) — eligible is an objective predicate which could be primarily based on the nature of *Agent* or on the nature of *Action* depending on the domain in which it is invoked.

Charlotte enters the interaction upon reception of the message advocate(alanna), and is such is accorded by Benjamin her own portrayal instance $\mathcal{P}$[charlotte]. This instance is identical to that of Benjamin's at the time of induction. Simultaneously, Alanna is informed of Charlotte's presence, ensuring that all future portrayal updates are duly

delivered to all three agents. Charlotte can now articulate her own arguments in response to the arguments already within the portrayal — for alacrity, let us assume that Charlotte finds all existing arguments to be at least admissible.

Alanna, Benjamin and Charlotte can now portray the constraints on Charlotte's role in interaction. Charlotte immediately makes the following arguments:

$$\begin{aligned}
\mathbf{k}' &= \langle\ \{\ \mathsf{controls}(\mathsf{charlotte}, \mathsf{permissions}(\mathsf{library}))\ \},\\
&\qquad \mathsf{controls}(\mathsf{charlotte}, \mathsf{permissions}(\mathsf{library}))\ \rangle\\
\mathbf{l}' &= \langle\ \{\ \mathsf{trusts}(\mathsf{charlotte}, \mathsf{benjamin})\ \},\ \mathsf{trusts}(\mathsf{charlotte}, \mathsf{benjamin})\ \rangle\\
\mathbf{m}' &= \langle\ \{\ \mathsf{certified}(\mathsf{alanna}),\\
&\qquad \neg\mathsf{at_capacity}(\mathsf{library}),\\
&\qquad \forall X, Y.\ \mathsf{certified}(X) \wedge \neg\mathsf{at_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))\ \},\\
&\quad \mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\ \rangle\\
\mathbf{n}' &= \langle\ \{\ \neg\mathsf{certified}(\mathsf{alanna})\ \},\ \neg\mathsf{certified}(\mathsf{alanna})\ \rangle\\
\mathbf{o}' &= \langle\ \{\ \mathsf{certified}(\mathsf{alanna})\ \},\ \mathsf{certified}(\mathsf{alanna})\ \rangle
\end{aligned}$$

Charlotte herself accepts arguments $\mathbf{k}'$ and $\mathbf{l}'$, but is undecided about argument $\mathbf{m}'$, being uncertain as to the truth of the proposition $\mathsf{certified}(\mathsf{alanna})$. This uncertainty is represented by the rebutting arguments $\mathbf{n}'$ and $\mathbf{o}'$.[3]

Alanna and Benjamin both accept arguments $\mathbf{k}'$ and $\mathbf{l}'$ without dispute. The two agents must also reconcile arguments $\mathbf{m}'$, $\mathbf{n}'$ and $\mathbf{o}'$ with their beliefs. From $\mathcal{C}_A$, Alanna can construct the following elaboration of $\mathbf{o}'$:

$$\begin{aligned}
\mathbf{o}'' &= \langle\ \{\ \mathsf{employed}(\mathsf{alanna}, \mathsf{edinburgh}),\\
&\qquad \mathsf{backing}(\mathsf{alanna}, \mathsf{edinburgh}),\\
&\qquad \forall X.\ \mathsf{employed}(X, Y) \wedge \mathsf{backing}(X, Y) \rightarrow \mathsf{certified}(X)\ \},\\
&\quad \mathsf{certified}(\mathsf{alanna})\ \rangle
\end{aligned}$$

This argument can also be integrated into $\mathbf{m}'$:

$$\begin{aligned}
\mathbf{m}'' &= \langle\ \{\ \mathsf{employed}(\mathsf{alanna}, \mathsf{edinburgh}),\\
&\qquad \mathsf{backing}(\mathsf{alanna}, \mathsf{edinburgh}),\\
&\qquad \forall X.\ \mathsf{employed}(X, Y) \wedge \mathsf{backing}(X, Y) \rightarrow \mathsf{certified}(X),\\
&\qquad \neg\mathsf{at_capacity}(\mathsf{library}),\\
&\qquad \forall X, Y.\ \mathsf{certified}(X) \wedge \neg\mathsf{at_capacity}(Y) \rightarrow \mathsf{eligible}(X, \mathsf{access}(Y))\ \},\\
&\quad \mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))\ \rangle
\end{aligned}$$

In response to this, Benjamin makes argument $\mathbf{p}'$:

$$\mathbf{p}' = \langle\ \{\ \neg\mathsf{backing}(\mathsf{alanna}, \mathsf{edinburgh})\ \},\ \neg\mathsf{backing}(\mathsf{alanna}, \mathsf{edinburgh})\ \rangle$$

[3]The obvious question is why we have a separate argument **o** rather than acknowledge that acceptance of argument **m** attacks **n**. This is a limitation of the particular formulation of assumption-based argumentation used in Chapter 3 — in particular the notion of attack used. Other formulations are possible which do not produce such extraneous arguments, at the cost of more complex definitions. The important thing to note is that the particular choice of assumption-based argumentation framework makes no difference to the portrayal mechanism as specified in Chapter 5, nor does it affect the critical definitions of synchronisation (4.8), sufficient expressivity (4.9) and reconciliation (4.10).

Assume that due to communication delays, Benjamin receives argument $\mathbf{o}''$ before $\mathbf{o}'$. In such a circumstance, Benjamin would insert $\mathbf{o}''$ into $\mathcal{P}[\mathsf{benjamin}]$ as a new argument rather than an elaboration, ensuring that Benjamin need do nothing upon finally receiving $\mathbf{o}'$ (see §5.4.2).

In any case, upon invoking reconcile, Charlotte finds that she can counter Benjamin's objection with a final argument $\mathbf{q}'$ — this is an example of an argument which Charlotte had not associated with arguments for $\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ before, having not previously linked $\mathsf{backing}(X,Y)$ with $\mathsf{certified}(X)$:

$$\begin{aligned}\mathbf{q}' = \langle\ \{\ &\mathsf{researcher}(\mathsf{alanna}),\\ &\mathsf{employed}(\mathsf{alanna},\mathsf{edinburgh}),\\ &\forall X,Y.\ \mathsf{researcher}(X) \wedge \mathsf{employed}(X,Y) \rightarrow \mathsf{backing}(X,Y)\ \},\\ &\mathsf{backing}(\mathsf{alanna},\mathsf{edinburgh})\ \rangle\end{aligned}$$

Assuming that this is sufficient to reconcile all agents' beliefs with portrayal $\mathcal{P}$, each updated portrayal instance will contain the following:

- $\Sigma = \{\mathsf{alanna}, \mathsf{benjamin}\}$.
- $\Upsilon = \{\neg\mathsf{accessible}(\mathsf{alanna},\mathsf{library}), \mathsf{patron}(X_1,\mathsf{library}), \mathsf{controller}(X_2,\mathsf{library}),$ $\mathsf{trustworthy}(\mathsf{alanna},\mathsf{access}(\mathsf{library})), \mathsf{controls}(\mathsf{charlotte},\mathsf{permissions}(\mathsf{library})),$ $\mathsf{trusts}(\mathsf{charlotte},\mathsf{benjamin}), \mathsf{eligible}(\mathsf{alanna},\mathsf{access}(\mathsf{library}))\}$.
- $\mathcal{A} = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}', \mathbf{e}'', \mathbf{f}', \mathbf{g}'', \mathbf{h}', \mathbf{i}', \mathbf{j}', \mathbf{k}', \mathbf{l}', \mathbf{m}'', \mathbf{n}', \mathbf{o}'', \mathbf{p}', \mathbf{q}'\}$, where $\mathbf{e}'' \rightharpoonup \mathbf{g}''$, $\mathbf{e}'' \rightharpoonup \mathbf{j}'$, $\mathbf{g}'' \rightharpoonup \mathbf{e}''$, $\mathbf{h}' \rightharpoonup \mathbf{e}''$, $\mathbf{h}' \rightharpoonup \mathbf{q}'$, $\mathbf{i}' \rightharpoonup \mathbf{g}''$, $\mathbf{i}' \rightharpoonup \mathbf{j}'$, $\mathbf{j}' \rightharpoonup \mathbf{e}''$, $\mathbf{n}' \rightharpoonup \mathbf{m}''$, $\mathbf{n}' \rightharpoonup \mathbf{o}''$, $\mathbf{o}'' \rightharpoonup \mathbf{n}'$, $\mathbf{p}' \rightharpoonup \mathbf{m}''$, $\mathbf{p}' \rightharpoonup \mathbf{o}''$ and $\mathbf{q}' \rightharpoonup \mathbf{p}'$.
- $\mathsf{inv}_{\mathcal{A}}(\mathsf{alanna} \mid \mathsf{benjamin} \mid \mathsf{charlotte}) = \emptyset$.
- $\mathsf{inv}_{\rightharpoonup}(\mathsf{alanna} \mid \mathsf{benjamin} \mid \mathsf{charlotte}) = \emptyset$.
- $\mathsf{obs}(\mathsf{alanna}) = \{\mathsf{researcher}(\mathsf{alanna})\}$ and $\mathsf{obs}(\mathsf{benjamin} \mid \mathsf{charlotte}) = \emptyset$.
- $\mathsf{acc}(\mathsf{alanna} \mid \mathsf{benjamin} \mid \mathsf{charlotte}) = \{\mathbf{a}', \mathbf{b}', \mathbf{c}', \mathbf{d}', \mathbf{e}'', \mathbf{f}', \mathbf{i}', \mathbf{m}'', \mathbf{o}'', \mathbf{q}'\}$.

In this case arguments $\mathbf{n}'$ and $\mathbf{o}''$ are equally admissible, but the agents have all decided to favour $\mathbf{o}''$ (probably because $\mathbf{o}''$ has more substantive reasoning behind it; $\mathbf{n}'$ is not conclusively defeated however). Charlotte thus unfolds her role in interaction as follows:

a(controller(library), charlotte) ::

c(advocate(alanna) $\Leftarrow$ a(advocate(library), benjamin)) then

c(permit(alanna, access(library))) then

c(permit $\Rightarrow$ a(applicant(library), alanna)).

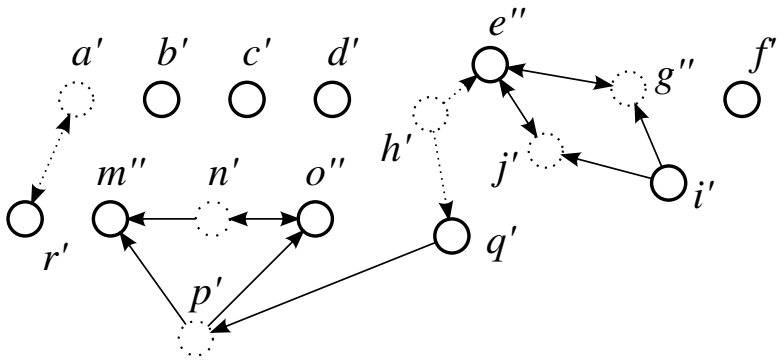


Figure 6.3: *The accepted interpretation of the system of arguments within the interaction portrayal of §6.1.4.*

Charlotte is required to perform the action $\mathsf{permit}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$. As one might expect, the effect of this action is to change the environment such that Alanna can access library. This will affect Alanna's observations. Thus Alanna will have to execute $\mathsf{observe}(\mathcal{P}[\mathsf{alanna}], \{\mathsf{accessible}(\mathsf{alanna}, \mathsf{library})\})$, which will result in the immediate dismissal of argument $\mathbf{a}'$ and the creation of argument $\mathbf{r}'$ in its stead, where $\mathbf{r}' \rightharpoonup \mathbf{a}'$ (and vice versa):

$$\mathbf{r}' = \langle\ \{\ \mathsf{accessible}(\mathsf{alanna}, \mathsf{library})\ \}, \mathsf{accessible}(\mathsf{alanna}, \mathsf{library})\ \rangle$$

The articulation of $\mathbf{r}'$ is dictated because $\mathbf{r}' \in \Delta$ (i.e. Alanna no longer rejects the claim of $\mathbf{r}'$, and $\mathbf{r}'$ supports a portrayable proposition in the portrayal). We can also expect that Charlotte will observe $\mathsf{accessible}(\mathsf{alanna}, \mathsf{library})$ independently, being clearly within her competence to do so (it being a consequence of her actions). This will affect the declared accepted arguments within portrayal $\mathcal{P}$, removing $\mathbf{a}'$ and inserting $\mathbf{r}'$.

Meanwhile, with the dispatch of message permit, control returns to Alanna. If Charlotte grants permission to Alanna, then Alanna should be able to confirm this by observation in her own role:

a(applicant(library), alanna) ::
    c(request ⇒ a(advocate(library), benjamin)) then
    succeed ← permit ⇐ a(controller(library), charlotte) ∧ accessible(alanna, library).

Confirmation is obtained by re-evaluating the proposition $\mathsf{accessible}(\mathsf{alanna}, \mathsf{library})$.

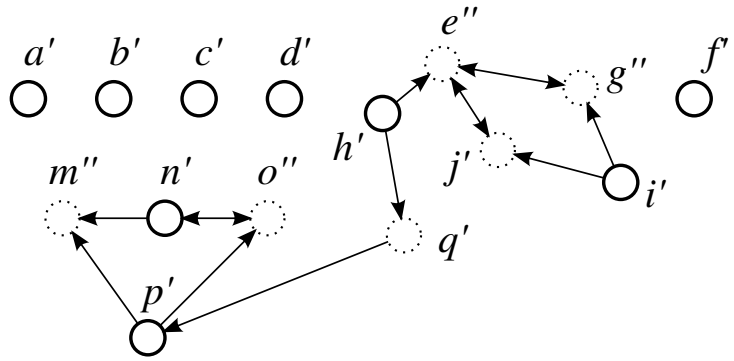


Figure 6.4: *An alternative accepted interpretation of the system of arguments within the interaction portrayal of §6.1.4 from the perspective of agent Charlotte.*

6.2 Alternative Outcomes

In the preceding description of interaction portrayal, the evaluation of the portrayal executed very cleanly, with a common consensus as to the end result. However a number of other (valid) outcomes could have transpired. For instance:

- If Charlotte had not accepted Alanna's observation that $\mathsf{researcher}(\mathsf{alanna})$, then argument $\mathbf{h}'$ would not have been dismissed, and not only would Charlotte have not accepted that $\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$, but Charlotte would also have had to reject $\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$ as well (unless additional arguments to defend argument $\mathbf{m}''$ could be constructed by any of the three available agents). The former would not be so bad for Alanna, because Benjamin would have already made his decision, and Charlotte had put her trust in Benjamin, but the latter would have led Charlotte to reject Alanna's application.

- Similarly, if Charlotte had chosen to accept $\mathbf{n}'$ instead of $\mathbf{o}''$, Charlotte would again have rejected Alanna's application.

- Alternatively, if Charlotte had been able to posit an argument defeating argument $\mathbf{e}''$ such that Alanna could not be determined as trustworthy based on the current evidence, this would *not* have necessarily affected the outcome of interaction (Benjamin having already made his commitment earlier), but could provide a basis for Benjamin to reject requests from Alanna in future, or even to begin a new interaction to retract Alanna's newly acquired access.

Another thing not illustrated in the prior example is invalid argumentation. All arguments posited above were complete insofar as the claim could be deduced from the supporting assumptions, which is quite reasonable for basic examples. In more complex cases, we might want to drop certain 'commonly known' domain rules from arguments, assuming that such rules are included in the logical frameworks of all agents [Hunter, 2007]. If such a presumption is in error however, there may be posited arguments which are uninterpretable by peers. For example, recall argument $\mathbf{g}''$:

$$\begin{aligned}\mathbf{g}'' = \langle\ \{\ & \exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X)),\\ & \forall X, Y.\ \mathsf{trustworthy}(\mathsf{X}, \mathsf{access}(\mathsf{Y})) \rightarrow \neg\exists Z.\ \mathsf{abused}(X, \mathsf{access}(Z))\ \},\\ & \forall X.\ \neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{X}))\ \rangle\end{aligned}$$

Perhaps Alanna (the positing agent) thinks that $\forall X, Y.\ \mathsf{trustworthy}(X, \mathsf{access}(Y)) \rightarrow \neg\exists Z.\ \mathsf{abused}(X, \mathsf{access}(Z))$ is common knowledge. Thus she might instead posit the following:

$$\begin{aligned}\mathbf{g}'' = \langle\ \{\ & \exists X.\ \mathsf{abused}(\mathsf{alanna}, \mathsf{access}(X))\ \},\\ & \forall X.\ \neg\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{X}))\ \rangle\end{aligned}$$

If Benjamin does not recognise the missing rule, then Benjamin would go on to invoke $\mathsf{dismiss}(\mathcal{P}[\mathsf{benjamin}], \mathbf{g}'')$ (§5.4.8), and add $\mathbf{g}''$ to $\mathsf{inv}_{\mathcal{A}}(\mathsf{benjamin})$. Benjamin would then *not* replace $\mathbf{g}'$ (which is valid to Benjamin) with elaboration $\mathbf{g}''$ and argument $\mathbf{g}'$ would not be attacked by $\mathbf{i}'$, and so $\mathbf{g}'$ would not be rejected from Benjamin's perspective. Benjamin would still attack $\mathbf{e}''$ with argument $\mathbf{j}'$ and notice that $\mathbf{i}'$ attacked $\mathbf{j}'$, but Benjamin would still reject argument $\mathbf{e}'$ on the basis of $\mathbf{g}'$. Alanna could however prevent this by invoking $\mathsf{expand}(\mathcal{P}[\mathsf{alanna}], \mathbf{g}'', \mathbf{g}''')$, which would replace $\mathbf{g}''$ with $\mathbf{g}'''$ (which would be identical to the original $\mathbf{g}''$) — this would lead to the same outcome described in the previous section.

An important question though is what would have happened in this scenario if no portrayal had been used. This can be determined quite easily by looking at the initial arguments posited by agents immediately after each expansion of the portrayal argument space. The first disputed proposition was $\mathsf{trustworthy}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$. Alanna posited argument $\mathbf{e}'$ for, and Benjamin posited argument $\mathbf{g}'$ against. The combination of arguments involving Alanna and Benjamin ultimately supported Alanna's trustworthiness. In this respect the use of a portrayal aided Alanna — without a portrayal Alanna would have been immediately rejected on the assumption that Alanna has abused trust in the past. Of course, the validity of Benjamin's revised belief depends on Alanna's sincerity; the portrayal mechanism offers no inherent protection

against deceptive peers, merely collating evidence. If Benjamin (as an intelligent, autonomous entity) had reason to doubt Alanna's testimony, then Benjamin could have drawn his conclusions in spite of the portrayal, but this is beyond the concerns of this thesis.

The other contentious proposition was $\mathsf{eligible}(\mathsf{alanna}, \mathsf{access}(\mathsf{library}))$. In this case, Charlotte illustrated her uncertainty with arguments $\mathbf{m}'$, $\mathbf{n}'$ and $\mathbf{o}'$, effectively inviting her peers to produce conclusive evidence. If we pretend that Benjamin did consider Alanna to be trustworthy even without the portrayal, then Alanna would still have met failure, because Charlotte would have been undecided about her eligibility and thus would have been forced to reject Alanna's application.

Naturally, the portrayal can go the other way; perhaps Benjamin and Charlotte could have been very receptive to Alanna's application, but upon using the portrayal to compare beliefs, they would have been led to reject Alanna.

A final consideration; what if instead of using the portrayal mechanism, we simply revised the acquire_access protocol to do what we just did with a portrayal? In answer, consider how we might revise the advocate role. In acquire_access, we specified the obligations of an advocate as so:

a(advocate(*Resource*), *Advocate*) ::
request ⇐ a(applicant(*Resource*), *Applicant*) then
(advocate(*Applicant*) ⇒ a(controller(*Resource*), *Controller*)
← controller(*Controller*, *Resource*) ∧ trustworthy(*Applicant*, access(*Resource*))
else
decline ⇒ a(applicant(*Resource*), *Applicant*)).

In order to replicate the dialogue that naturally arises from the interaction portrayal of §6.1.3, we would have to specify advocate as so:

a(advocate(*Resource*), *Advocate*) ::
request ⇐ a(applicant(*Resource*), *Applicant*) then
(query(researcher) ⇒ a(applicant(*Resource*), *Applicant*)
← ¬ beneficial(*Applicant*, abuse(access(*Resource*))) then
a(interrogator(*Actions*, access(*Resource*), *Applicant*), *Advocate*)
← potential_abuses(*Applicant*, *Actions*) then
(advocate(*Applicant*) ⇒ a(controller(*Resource*), C)
← pass ⇐ a(interrogator(_, access(*Resource*), *Applicant*), *Advocate*)
else
decline ⇒ a(applicant(*Resource*), *Applicant*)

```
        ← fail ⇐ a(interrogator(_, access(Resource), Applicant), Advocate)   )
    else
    decline ⇒ a(applicant(Resource), Applicant)   ) ← controller(C, Resource).
```

This role model sacrifices the generality of the trustworthy predicate, instead determining specifically whether there is any benefit in the applicant abusing access to the given resource before checking to see whether the applicant is a researcher. It then demands that the advocate assume a new role as interrogator in order to check for prior abuses by the applicant which might undermine the presumption of the applicant's trustworthiness. This means that we also have to specify the role of interrogator. We can define a base case where there are no prior actions to check:

```
a(interrogator(∅, access(Resource), _), Interrogator) ::
    pass ⇒ a(advocate(Resource), Interrogator).
```

We also define a step case where the interrogator challenges the defender (the applicant in this scenario) to provide an explanation for any analogous acts which might demonstrate a lack of trustworthiness as regards accessing the particular resource in question:

```
a(interrogator([ Action | Actions ], access(Resource), Defender), Interrogator) ::
    challenge(Action) ⇒ a(defender, Defender)
    ← analogous(Action, access(Resource)) then
    response(Counter) ⇐ a(defender, Defender) then
    (   a(interrogator(Actions, access(Resource), Defender), Interrogator)
        ← ¬ applicable(Action, access(Resource), Counter)
        else
        fail ⇒ a(advocate(Resource), Interrogator)   ).
```

If the defender is unable to explain away any given action, then the interrogator will decline to advocate it. Of course, we also need a specification for defender, which actually tries to produce such explanations:

```
a(defender, Defender) ::
    challenge(Action) ⇐ a(interrogator(Actions, Comparison, Defender),
                          Interrogator) then
    (   response(Counter) ⇒ a(interrogator(Actions, Comparison, Defender),
                              Interrogator)
        ← ¬ applicable(Action, Comparison, Counter)
        else
```

response(null) ⇒ a(interrogator(*Actions*, *Comparison*, *Defender*),
Interrogator)).

These role models (in combination with the other parts of acquire_access) are significantly more complex than the simple advocate model used in tandem with the portrayal mechanism. Furthermore, they demonstrate additional problems:

- The revised models are more domain-specific than the generic model. They assume that trustworthiness is based on the applicant's status as a researcher (perhaps valid for a library, but not for other resources) and a lack of detectable benefit in abusing trust. They also assume that this trustworthiness is undermined by analogous instances of abuse.

- The revised models only provide a single procedure for determining trustworthiness. The applicant has to confirm their status as a researcher, but the advocate decides independently whether or not there is any benefit in the applicant abusing trust. The applicant then has to defend itself as regards past behaviour that the advocate is aware of. It is possible to change the burden of proof of course, but any given configuration will trade one set of problems for another.

- Despite being a more explicit protocol for the specific problem case described in previous sections, the actual constraints imposed are *less* clear than the generic originals. Whilst a predicate like trustworthy is both abstract and subjective, it is easier to identify what the predicate is evaluating than a predicate like applicable, which here is used to compare one action to another in light of a given 'counter' — only in a system with a very well-defined single ontology can such a predicate be used with any reasonable expectation that agents will be able to correctly interpret any resulting proposition.

This illustrates the *other* way by which we might consider the contribution of this thesis. On the one hand, we have a mechanism by which interactions can be augmented with additional unstructured dialogue based on a well-defined formal model (the synchronisation of beliefs within a shared argument space). On the other hand, we have a justification for simpler, more generically-applicable interaction protocols.

6.3 Implementation Requirements

Interaction portrayals operate in the intersection of multi-agent coordination and agent-based defeasible reasoning. Any implementation of the portrayal mechanism specified in Chapter 5 is wholly dependent on the implementation of these two things. In the case of multi-agent coordination, a functional system for deploying and executing agent interaction protocols in a distributed environment is required. In the case of agent-based defeasible reasoning, for each agent a knowledge base and an inference engine capable of abductive (and perhaps inductive) reasoning is required; this machinery must then be comprehensible as an argumentation process such that we can then extract concrete arguments which can then be mapped into the argument space of a portrayal. In the context of these systems, the portrayal mechanism itself is relatively simple, its theoretic properties being the significant point of interest (hence the focus of this thesis).

The deployment and execution of agent interactions is perhaps the simplest component of an implementation to deal with, insofar as existing systems can be exploited. For LCC, the Open Knowledge system [Siebes et al., 2007] can be used, for example. For simulation purposes, an LCC interpreter [Robertson, 2004, Robertson et al., 2008] can be implemented easily in any programming language (Prolog is particularly suitable). Any simulated environment must support distinct autonomous agents (or abstractions thereof), and must support extensions to the core coordination mechanism to support the inclusion of the portrayal mechanism (this would be the most obvious issue with using an off-the-shelf platform).

The modelling of agents themselves may be more difficult. Interaction portrayals are conceived to assist in scenarios where agents possess a significant depth of knowledge in various different domains, and where there needs to be spontaneous ability to produce an efficient abstraction of that knowledge during interaction. Thus any non-trivial implementation of the portrayal mechanism requires agents with non-trivial data corpuses. In order to produce the potential arguments necessary for producing a portrayal, these agents require access to efficient algorithms for abducting new hypotheses and then interpreting the consequences of those hypotheses given prior observations — basically, fully functional argumentation systems. If merely simulating a multi-agent system, some concern for the overall practical efficacy of the resulting system of complex agents will also be necessary.

The result of any implementation would be to give additional confidence in the

inherent executability of the portrayal mechanism under a variety of circumstances — in particular, if the defeasible reasoning processes of individual agents in a multi-agent system are capable of generating arguments in a real-time system, then the portrayal mechanism should be able to produce potential arguments in support of agents' beliefs during interaction. Of course the converse also applies — the portrayal mechanism will not magically make it feasible to reason at a level of expressivity beyond what an agent is competent to process. Thus, a direction of further work would be to categorise the logics best suited for an agent to map their beliefs into in the context of an interaction portrayal (as mentioned in §7.2).

Chapter 7

Conclusions

In this thesis, we explored the use of argumentation as a means to discuss constraints imposed on multi-agent interaction by an interaction protocol *during* interaction. In particular, we have examined how dialogue between heterogeneous agents with very different beliefs can be constrained in such a way as to minimise computation whilst still making adequate use of the distributed knowledge available to peers. This has led us to specify a distributed logical mechanism by which constraints on interaction can be resolved based on collaborative reasoning — a mechanism for *portraying* interactions which ensures various desirable properties defined for this purpose. With this mechanism we essentially provide a means for opportunistic, prioritised belief revision between heterogeneous agents on demand, wherein beliefs are prioritised based on how they are seen to bear influence on the outcomes of interactions between those agents. Thus, this thesis can be distinguished from prior research in two aspects:

- The use of potential argumentation and the reconciliation of argument systems bridges the gap between the private formulation of beliefs on the part of individual agents and the mapping of arguments representing those beliefs into a social argument space.

- The portrayal mechanism operates during an active interaction, responding to developments in the interaction state, without requiring any augmentation of the protocol to which a portrayed interaction adheres.

In this final chapter, we overview the most notable elements of this thesis' contribution (§7.1) and ruminate on possible directions for future research (§7.2).

7.1 Discussion

Basically, what we desired was a process which could be executed during an interaction which would allow agents to discuss constraints imposed on that interaction before their resolution, so as to avoid outcomes not justified by the state of the system. Such outcomes might occur because agents are in possession of incomplete information, or have independently drawn false conclusions from what information is available. If agents are given motivation to posit claims and arguments during interaction such that their peers can then correct any perceived fallacies in those claims or arguments, then it can reasonably be expected that the end conclusion will more likely match objective reality (where objective constraints are to be evaluated) or communal wisdom (for more subjective constraints). We wanted to do this without simply relying on replacing existing interaction protocols with more complex, over-specified variants which then lose all general applicability.

By relying on the resolution of a distributed argumentation problem, one which adapts to the state of the concurrent interaction with which it is associated, we are able to generate dialogues discussing constraints and their expected resolutions even for interactions based on generic, unaugmented protocols. In addition, by allowing the argumentation process to interface directly with the beliefs of agents, we effectively perform distributed belief maintenance based on the logical propositions which must be evaluated in order to resolve those interaction constraints.

In Chapter 2 we formalised the notion of distributed interaction so as to ensure that we had something concrete to augment with our main contribution:

A concrete formalism for interaction. In order to counteract a perceived vagueness in artificial agent literature, we provided a formal specification for an arbitrary interaction and interaction dialogue in §2.1; this was made use of in Chapter 5.

An abstract specification of interaction modelling. In §2.2 we defined what we believed to be the essential forms of an interaction model and an interaction state, and we described how they related to one another. This provided us a basis on which to analyse past agent interaction literature, as well as the ability to formally state the nature of the problem this thesis attempts to deal with.

A process model for distributed interaction. In §2.3 we specified a logical process model for distributed interaction. This allowed us to identify precisely how the portrayal mechanism of Chapter 5 interfaced with the interaction being por-

trayed. This particular model of the interactive process was an adaptation of the model of linearised peer interaction described in [Robertson et al., 2008]. Its usefulness stemmed from being concrete enough to identify various important facets of interaction (initialisation, role adoption, constraint resolution, etc.), whilst still being abstract enough to be applicable to a number of different implemented systems (distributed versus centralised, pre-determined agent groups versus *ad-hoc* groupings, etc.).

At the core of our contribution is the idea that social argumentation is primarily a medium through which arguments can be mapped from one private argumentation framework to another. In particular, by allowing agents to consider a shared argumentation process with respect to another (possibly more complex) argumentation process (as embodied by the theory contexts of §3.2.4), we allow agents to draw conclusions about the argument space in which the shared process is conducted — this is not possible if one considers an argumentation framework in isolation. We are thus able to infer whether or not the arguments generated within a shared argument space are balanced (Definition 4.6), whether or not the shared space is sufficiently expressive (Definition 4.9) and whether or not the arguments within are reconciled with those held privately by an observer (Definiton 4.10). Ultimately, we can determine whether or not a set of theories is *synchronised* within a given argument space (Definition 4.8). Practically speaking, this allows us to ensure that decisions made within that space are admissible to all peers. Specifically, in Chapter 3 we made the following contributions:

A formal notion of argument space. In §3.2.2 we introduced into this thesis the notion of an argument space, which limits the scope of an argumentation framework by restricting the system of arguments which could be generated within that framework. This notion underpins our contribution by allowing us to consider the construction of arguments in different argument spaces and ultimately the transference of arguments between spaces. Whilst argument spaces have always been tacit in all argumentation frameworks described in the literature, we have not seen elsewhere argument spaces described explicitly as a component of an argumentation framework in quite the same form we use here.

An argumentation-based context for agent theories. In §3.2.4 we provided a description of the context of a theory produced using assumption-based argumentation. By drawing the components of such a context together we are able to provide a succinct description of the basis for the production (and revision) of

a theory which allows us to directly map arguments between an agent's theory context and a shared argument system, allowing us to define relationships between an agent's beliefs and the state of a social argumentation process more easily in terms of potential restrictions (or expansions) into (or out of) different argument spaces.

A relational notion of potential argument. In §4.1.1 we used the notion of a potential argument as the basis for mapping arguments from one argument space into another, more restricted space. We also looked briefly at the reverse case. Such potential argumentation serves as an implementation-agnostic way to determine the detail required of arguments articulated within a portrayal as the argument space of that portrayal is refined.

A notion of argument mapping between contexts. Elaborating upon the prior point, the notion of argument potential allowed us to describe how sets of arguments can be mapped into a more restrictive or permissive argument space; we considered both potential restrictions of arguments, and potential expansions. More generically, this can be seen as a basis for the general mapping of arguments from one context to another, which we believe to be useful for abstracting or combining information from different reasoning architectures.

A notion of belief synchronisation. In §4.2 we introduced the notion of belief synchronisation within a given argument space as a basic goal state to which a social argumentation process can aspire. In essence, synchronisation ensures that any claim made by an agent within the chosen argument space will be admissible to all agents given their own beliefs and observations. This is in lieu of being able to enforce a single common interpretation of arguments on all peers — instead, we content ourselves with agents making decisions compatible with the available evidence. By limiting the 'available evidence' to a given argument space, we can limit computation to a reasonable portion of agents' knowledge bases, rather than attempt an intensive process of truth maintenance over multiple (large) belief stores.

A notion of sufficient expressivity in argument spaces. It is important that the argument space in which synchronisation is tested is useful. An argument space is *sufficiently expressive* with regards to a given agent's theory if any attacks against that theory which the agent can counter privately it can counter within that ar-

gument space. As long as the argument space remains sufficiently expressive, then it is possible to to synchronise all agents beliefs within that space. It has been shown in §5.1.3 that a portrayal's argument space will always accept new arguments such that it becomes sufficiently expressive, even whilst otherwise attempting to minimise the space.

A notion of argument balance. An alternative approach to ensuring a useful argument space is to check whether or not the arguments which can be generated within that space are *balanced* (as per Definition 4.6 at the end of §4.1.2). A system of arguments is balanced with respect to an outside context if any common premises deemed necessary to ultimately support arguments in the argument system are made explicit in all such arguments if already explicit in one. The practical effect of such balancing is to ensure that the base premises on which any accepted extension of arguments is supported are as independent of one another as possible, such that any derived theory is a 'good' basis for deriving conclusions about its subject.

A notion of reconciliation between argument systems. The system of arguments in an agent's theory context is reconciled with another system of arguments in a different context if the other argument system not only reflects the beliefs of the agent (such that there exists an admissible[1] extension of that system which is a potential restriction of the accepted extension in the agent's theory context), but every admissible extension of the other argument system is admissible within the theory context. It has been shown in §4.2 that if every agent involved in argumentation is able to reconcile their theory contexts with the system of arguments generated, then the beliefs of those agents will be synchronised within that system's argument space. Thus reconciliation can be used to achieve synchronisation without requiring knowledge of every peer's beliefs. This is especially important for portrayal-augmented interactions which must synchronise agent beliefs within the argument space of a portrayal.

Having established a notion of a social argument space into which agents could map arguments constructed privately within their own theory contexts and *vice versa*, and having established the criteria by which the quality of such a space could be evaluated, it then became necessary for us to put theory into practice.

[1] In the sense of Definition 3.8.

An interaction portrayal stored the shared system of arguments generated in discussing constraints imposed on an interaction. Using local knowledge of the interaction state and the arguments already posited, agents could determine how to map their beliefs into the portrayal. After any necessary belief revision brought about by the arguments of peers, those agents could then be assured that their beliefs are synchronised within the argument space of the portrayal. That argument space changes with the interaction state, so perhaps necessitating further discussion. In Chapter 5 we provided the following:

A formal decision problem for constraint discussion. The portrayal mechanism incrementally constructs an argument space for social argumentation in which agents can synchronise their beliefs. This argument space is focused on the constraints imposed on interaction by its protocol, and is expanded as necessary to sufficiently describe the conflicts which exist between agents' favoured interpretations of the evidence available to each of them. During the process of reconciliation, agents assimilate new information and defend their (revised) beliefs, driving the construction of an interaction portrayal and ensuring that all peers account for the information posited in that portrayal.

A mechanism which operates alongside interaction. The portrayal mechanism executes during an augmented interaction. There is no need to modify any interaction protocol in order to ensure compatibility with interaction portrayals as long as the logical propositions constituting constraints on interaction can be clearly identified. This distinguishes this work from one in which agents simply argue about a set of propositions in isolation.

A mechanism which adapts with the interaction state. As interaction develops, so does its portrayal. It cannot necessarily be determined in advance which constraints out of those which can be extracted from a protocol will apply to the specific interaction about to unfold. It may also be inefficient to argue about certain propositions before certain decisions are made (particularly where as-yet uninstantiated variables are concerned). By portraying an interaction during its execution, we can resolve these issues.

A mechanism which operates in a dynamic environment. In §3.2.3 we were able to provide a simple justification for the use of argumentation in dynamic domains. By noting the lack of practical difference between views of a state and

multiple states, and by permitting the temporary dismissal of arguments which contradict observations of the environment, we were able to quickly show that it is feasible to describe an artefact of many states using a single stable argument system (as per Definition 5.14). Such a system can simultaneously describe the different states of the artefact as different admissible extensions of the system — extensions which can be 'collapsed' by observation of the environment.

The consequence of this with respect to our portrayal mechanism is that we need not concern ourselves with changes to the environment during an interaction. If the state of the world changes such that it impinges upon the portrayal argument system, then agents can simply observe the contradiction between the world state and existing arguments, and posit those alternative arguments which naturally arise under the new state. If the world state should revert, then past observations can be retracted, and new observations can be made to suppress now unacceptable arguments. In this fashion is the integrity of claims made during interaction sustained even in a dynamic system.[2]

A completely distributed mechanism. By choosing to distribute a portrayal amongst peers such that every agent involved in an interaction has its own portrayal instance, and by multi-casting each update message to all peers with such instances, we ensure that our mechanism is robust and suitable for environments where portrayed interactions are themselves fully decentralised.

Practical argumentation in a minimal space. It is difficult to determine in advance how large an argument space for social argumentation should be without actually sharing information between peers. As such, we have taken a policy of minimalism, permitting expansion where necessary to illustrate where conflicts exist between admissible claims. Such expansion ensures that the portrayal argument space always becomes sufficiently expressive (which ensures that synchronisation within that space is feasible). Expansion is conducted in a disciplined fashion, so as to keep the arguments in a portrayal as small as possible.

A mechanism which ensures reconciliation. The portrayal mechanism operates by requiring agents to respond to certain events under certain conditions; these re-

[2] It should be noted that this only has bearing to the interpretation of the portrayal, and not on decisions already made during interaction as prescribed by the interaction protocol, which agents have already executed. If the environment changes such that a past commitment is untenable, then agents can only respond as permitted by the protocol, though it may be possible in limited circumstances to backtrack [Osman, 2003].

sponses then generate messages to which their peers can respond. It has been shown that the invocation of operations constituting the portrayal mechanism will lead an agent to the reconciliation of its theory context with the portrayal argument system. As long as all agents act according to specification, we can thus ensure synchronisation with the portrayal argument space. If the portrayal space changes, then agents will re-reconcile.

Opportunistic belief revision on demand. Whilst superficially one might consider an interaction portrayal to be the artefact by which decisions are made during interaction, the reality is that it is merely a vessel for influencing the private beliefs of agents — agents still make decisions based on their beliefs and their beliefs alone. As mentioned many times prior to this, the effect of portraying an interaction is to perform a kind of distributed truth maintenance procedure across the set of agents involved in an interaction, focused only on those beliefs which can be seen to have bearing on the resolution of interaction constraints. Thus, what we have is a partial solution to the problem of how to reconcile inconsistencies between agent beliefs where attaining joint consistency is computationally intractable — we focus only on what is immediately important, and we decide this based on the interactions that agents choose to engage in.

A demonstration of potential argumentation in a practical system. Last but not at all least, the logical specification of the portrayal mechanism as a means to discuss constraints on interaction acts an example of potential argumentation and argument mapping put into practical use. The idea of mapping arguments between different contexts, and of combining the beliefs of agents within a limited space (so as to abstract aside irrelevant details) is applied here to the task of ensuring that decisions made during interaction by agents are made with the best information which can be practically made available given computational constraints.

Insomuch as this section has described our achievements, it now befalls us to discuss the work undone.

7.2 Future Work

Despite the contributions described in the previous section, there still exist a number of possible ways to expand upon this thesis. Broadly speaking, we can further develop

upon potential argumentation and the mapping of arguments from one framework to another, we can refine the portrayal mechanism, or we can perform further empirical experiments in order to better demonstrate how theory translates into practice.

Experimentation would primarily concern itself with the influence of portrayals over an extended period, or with measuring the usefulness of portrayals given different logics:

Evaluate systems of interaction rather than individual interactions. In this thesis we have limited ourselves to considering the effect of portraying interactions in isolation. We have not engaged in a significant evaluation of the influence of portrayals across multiple interactions over an extended period. There are a number of directions we could take such evaluation:

- We can seed an agent system with a mix of false and true information, and see if portrayals accelerate the process of purging the false information through discussion motivated by interaction.
- We can distribute tasks amongst agents which require interaction and provide only incomplete information to each peer such that we can then measure if portraying interactions allows agents to more quickly attain their goal states.
- We can provide a group of agents with a task which requires information which is not available to them; we can then test the abductive capabilities of agents and test the ability of portrayals to rectify inconsistencies between abductions at the point of interaction.

In any case, care must be taken as to the construction of experiments such that they are representative of real systems. The great difficulty of reasoning-driven agent systems research is the production of exemplar systems which possess the qualities and complexity of the kind of system the research is envisaged to assist. It is uncertain as to whether the toy problems often used in the literature to evaluate new models truly model average scenarios in many cases.

Evaluate interaction portrayals which use different logics. Whilst our treatment of argumentation and potential argumentation has been kept mostly abstract, examples have focused on propositional or first-order predicate calculus. There are of course other logical formalisms, such as description logics and higher order logics on two different extremes. It would be instructive to perform further

experimentation on the use of portrayals in domains described by such logics. Such experimentation would allow us to better evaluate whether portrayals are more suitable for more expressive logics for instance, or whether the value of portrayals is invariant to the complexity of the underlying logic and more linked to the volume of information expressed in the logic.

In tandem with further work into mapping arguments between different argumentation frameworks with different underlying logics, it would also be instructive to consider the use of portrayals where agents privately use different logics for internal reasoning. What interim logical formalism should a portrayal use? Is it more important that portrayal logic be at least as expressive as the logics used privately by agents, or that portrayal logic be no more expressive than the least expressive logic formalism used by a peer in an interaction?

Study the propagation of beliefs in multi-agent systems. One possibility we entertained in the first chapter was the idea that portrayal of multiple concurrent interactions would permit the transfer of information between interactions via a shared peer, such that the portrayal of one interaction would almost incidentally influence the outcome of the other interaction. It was felt that such unsolicited information dissemination was indicative of the strengthening of societal bonds within a particular agent group. Of great interest would be if there was some possibility of deriving a calculus of unsolicited information flow through an interacting agent system, such that we could better understand how dialogue produces a ‘common culture’ of assumptions amongst agents.

Alternatively, we can consider the process model for portrayal construction and maintenance, and consider how to make it more useful in a wider range of actual multi-agent systems:

Make the portrayal mechanism more robust. As specified in Chapter 5, the portrayal mechanism is distributed and asynchronous — insomuch as it considers the possibility of messages arriving out of order. The specification does not however consider the possibility of messages (or agents) being lost or of responses timing out. To a certain extent, this is the province of specific implementations and in particular of the networking protocol used underneath the logical layer at which our contribution operates. On the other hand, there does need to be contingencies in place for when individual agents become unable to continue

their contribution to a portrayal if their peers are to be able to continue (and finish) the interaction by themselves (of course, the primary interaction has to be completable without the missing agent, which is hardly a given).

Another thing to note is that we have assumed a mesh topology for agents, wherein every agent can be (and is) contacted by every other agent. It may be that other topologies may be more efficient in different circumstances. For example, a ring topology (where messages are passed around from one agent to another until they reach their originator) would remove the issue of mis-ordered messages, but would present an alternative issue of being very sensitive to agent failure.

Account for malicious or incompetent agents. It is assumed throughout this thesis that the agents engaged in an interaction are sincere and competent, such that any arguments made by an agent represent genuine considerations on its part and are posited into a portrayal in accordance with the rules given by the portrayal specification. Of course, this might not actually be the case. In question then, is how to modify the portrayal mechanism such that agents are unable or at least unwilling to act treacherously. Undoubtedly, this would require an additional level of verification and validation to be built into the portrayal mechanism which would increase complexity noticeably. In particular, there would need to be a means by which agents can identify an unruly peer and dismiss it from the portrayal (if not necessarily the interaction to which a portrayal is attached), so that its (faulty) contribution need no longer be considered.

Identify socially optimal interpretations of portrayal arguments. Throughout this thesis we have made the case that peers cannot dictate to a given agent its own beliefs, and thus if multiple admissible interpretations of a given portrayal argument system exist, then there is no justifiable basis on which to force communal selection of any one of those interpretations. Whilst we stand by this case, it is still possible that it may be advantageous, given no better reason to choose one interpretation over another, for an agent to submit to peer pressure in certain circumstances. In [Rahwan and Larson, 2008], a study is made of ‘socially optimal’ interpretation of argument systems — perhaps there are grounds for an agent to (voluntarily) favour a socially optimal interpretation over another when taking decisions in the interaction to which a portrayal is attached. This does not necessarily commit it to accepting that interpretation privately outside the

interaction though.

Finally, we can step back from concerning ourselves with interaction portrayals themselves, and further explore the underlying notions of potential argumentation and argument mapping applied by the portrayal mechanism:

Explore argument mappings between more different contexts. The foremost concern when mapping arguments between different contexts in this thesis has been with different argument spaces, with only limited attention given to different underlying logical frameworks. What attention that has been given has been focused on to what extent certain facts and rules can be subsumed by the framework. It would be of great interest to consider how arguments can be abstracted or restructured to fit into alternative contexts, and to consider how an argument admissible in one context may justifiably be inadmissible in another (or *vice versa*).

Study optimal argument spaces in different circumstances. The argument space in which an argumentation framework produces arguments determines which conflicting hypotheses an argumentation process will explore. In isolation, there is little we can say about the quality of a given argument space, except perhaps whether or not it describes a *flat* framework [Bondarenko et al., 1997]. Given an outside context however, we can draw further conclusions. We have already considered whether a given argument space is sufficiently expressive given an agent's standing beliefs, and we have considered whether the system of arguments which can be generated within a given argument space under a given logical framework is balanced. Other qualities which might be required of a 'good' argument space may include the ability to prove that peers will *not* be able to infer certain things from the mapping of arguments into a restricted space (which might be useful where sensitive information is concerned). More work could also be done to permit us a more complete understanding of *when* a particular property is important for argument spaces under what conditions.

Research an improved notion of argument refinement. For the most part, in this thesis we have considered arguments abstractly. Even where we look at the internal structure of arguments, we treat logical sentences as base assumptions or claims. Thus, if a given sentence is unacceptable as a whole, we reject the sentence entirely, and any variant of a given sentence is treated as a wholly different sentence. In practice, this is a rather clumsy approach to assumption-based

argumentation which forces the propagation of many similar arguments when debatably the more pragmatic approach would be to refine an existing argument. In [Carbogim, 2001], a number of schemata are specified for the modification of existing sentences in a theory in response to different attacks against that theory — an interesting project on the implementational side of this thesis would be to integrate such schemata into the workings of the portrayal mechanism such that a more subtle refinement of the portrayal space is then engaged in. This would have the effect of producing smaller argument spaces by permitting a finer grain of reconciliation between the portrayal and each agent's theory context.

Consider the opportunistic merging of interactions by peers. It has been assumed that interactions are instigated by a single agent. It is possible however for agents to instigate separate interactions, with distinct protocols, which might then be merged into a single interaction upon convergence of interests. A simple example would be one where there exists a server agent and a client agent — the client agent initiates a new interaction requesting a service from the server, whilst the server might be engaged in another interaction which is driven by such service requests. Whilst these interactions can be kept separate, it is possible that upon receiving a request, the protocols for both interactions could be merged by the server — consider the benefits of the client needing only a vestigial description of the server role, and the server only needing a vestigial description of the client role, with a combined protocol describing the 'true' interaction. In such a case, both agents will have conceived portrayals independently, and these portrayals will need to be merged along with their parent interactions.

It is apparent then that there is still work to be done if the portrayal mechanism, or something very like it, is to be deployed in a real-world system. Nevertheless, even if interaction portrayals are *not* ever implemented in the form described in this thesis, the concepts underpinning them remain generically useful as a basis by which to understand how dialogue between agents creates a social argument space which exists in the intersection of agents' private belief models. This is important for understanding how heteorogeneous information sources interact at the level of inference rather than merely ontology, and is itself a basis for understanding how abstractions of knowledge domains can be constructed collectively by peers with access to different types and levels of expertise (and of course, possibly conflicting beliefs which must be reconciled).

Returning specifically to the contribution of this thesis however, what we have

demonstrated is that using argumentation to discuss an interaction during its own execution is a viable strategy for improving the ability of intelligent agents to actually use that intelligence in dialogue. Agents can engage in interaction according to concise protocols with easily recognisable results and still engage in sophisticated debate without sacrificing basic executional feasibility or relinquishing their autonomy to over-elaborate dialogue models. Such debate allows for agents to disseminate knowledge and test their beliefs against those of peers, which should then allow agents to make more informed decisions based on the collective wisdom of peers, rather than simply with whatever knowledge they happened to have or not have prior to interaction. Intuitively, more informed decisions should lead to better outcomes for an interaction with respect to the intended purpose of that interaction.

Bibliography

[Austin, 1962] Austin, J. L. (1962). *How to Do Things with Words*. Oxford University Press.

[Black and Hunter, 2007] Black, E. and Hunter, A. (2007). A generative inquiry dialogue system. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems*, pages 1010–1017.

[Black and Hunter, 2008] Black, E. and Hunter, A. (2008). Using enthymemes in an inquiry dialogue system. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 437–444.

[Bondarenko et al., 1997] Bondarenko, A., Dung, P. M., Kowalski, R. A., and Toni, F. (1997). An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93(1–2):63–101.

[Bondarenko et al., 1993] Bondarenko, A., Toni, F., and Kowalski, R. (1993). An assumption-based framework for non-monotonic reasoning. In *Proceedings of the Second International Workshop on Logic Programming and Nonmonotonic Reasoning*, pages 171–189.

[Breiter and Sadek, 1996] Breiter, P. and Sadek, M. D. (1996). A rational agent as a kernel of a cooperative dialogue system: Implementing a logical theory of interaction. In *Proceedings of the ECAI-96 Workshop on Agent Theories, Architectures and Languages*, pages 261–276. Springer-Verlag.

[Caminada, 2006a] Caminada, M. (2006a). On the issue of reinstatement in argumentation. In *Proceedings of the 1st International Conference on Computational Models of Argument*, pages 121–130.

[Caminada, 2006b] Caminada, M. (2006b). Semi-stable semantics. In *Computational Models of Argument: Proceedings of COMMA 2006*, pages 121–130.

[Caminada, 2007] Caminada, M. (2007). Comparing two unique extension semantics for formal argumentation: Ideal and eager. In *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence*, pages 81–87.

[Carbogim, 2001] Carbogim, D. (2001). *Dynamics in Formal Argumentation*. PhD thesis, School of Informatics, University of Edinburgh.

[Davis and Smith, 1983] Davis, R. and Smith, R. G. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109.

[Dimopoulos et al., 1999] Dimopoulos, Y., Nebel, B., and Toni, F. (1999). Preferred arguments are harder to compute then stable extensions. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*.

[Doyle, 1979] Doyle, J. (1979). A truth maintenance system. *Artificial Intelligence*, 12(3):231–272.

[Dung, 1995] Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357.

[Dung et al., 2007] Dung, P. M., Mancarella, P., and Toni, F. (2007). Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10–15):642–674.

[Dunne, 2008] Dunne, P. (2008). The computational complexity of ideal semantics i: Abstract argumentation frameworks. In *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA 2008)*, pages 147–158.

[Esteva et al., 2002] Esteva, M., de la Cruz, D., and Sierra, C. (2002). Islander: an electronic institutions editor. In *Proceedings of the First Interactional Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 1045–1052.

[Esteva et al., 2001] Esteva, M., Rodrígeuz-Aguilar, J. A., Sierra, C., Garcia, P., and Arcos, J. L. (2001). On the formal specification of electronic institutions. In *Agent Mediated Electronic Commerce, The European AgentLink Perspective*, pages 126–147. Springer-Verlag.

[Esteva et al., 2004] Esteva, M., Rodríguez-Aguilar, J. A., Rosell, B., and Arcos, J. L. (2004). Ameli: An agent-based middleware for electronic institutions. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 236–243.

[Fagin et al., 1995] Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning About Knowledge*. The MIT Press.

[Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. In *CIKM '94: Proceedings of the third international conference on information and knowledge management*, pages 456–463. ACM Press.

[Flores and Kremer, 2002] Flores, R. A. and Kremer, R. C. (2002). To commit or not to commit: Modelling agent conversations for action. *Computational Intelligence*, 18(2):120–173.

[Gaertner and Toni, 2008] Gaertner, D. and Toni, F. (2008). Hybrid argumentation and its properties. In *Proceedings of the 2nd International Conference on Computational Models of Argument (COMMA 2008)*, pages 183–195.

[García and Simari, 2004] García, A. J. and Simari, G. R. (2004). Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138.

[García-Camino et al., 2005] García-Camino, A., Noriega, P., and Rodríguez-Aguilar, J. A. (2005). Implementing norms in electronic institutions. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 667–673.

[Genesereth and Fikes, 1992] Genesereth, M. R. and Fikes, R. E. (1992). Knowledge interchange format, version 3.0 reference manual, technical report logic-92-1. Technical report, Computer Science Department, Stanford University.

[Halpern and Moses, 1990] Halpern, J. Y. and Moses, Y. (1990). Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587.

[Hansson, 2003] Hansson, S. O. (2003). Ten philosophical problems in belief revision. *Journal of Logic and Computation*, 13:37–49.

[Huhns and Bridgeland, 1991] Huhns, M. N. and Bridgeland, D. M. (1991). Multiagent truth maintenance. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1437–1445.

[Huhns and Stephens, 2000] Huhns, M. N. and Stephens, L. M. (2000). Multiagent systems and societies of agents. In Weiss, G., editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 2. The MIT Press.

[Hunter, 2007] Hunter, A. (2007). Real arguments are approximate arguments. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, pages 66–71.

[Jennings, 1993] Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 2(3):223–250.

[Kakas et al., 1998] Kakas, A. C., Kowalski, R. A., and Toni, F. (1998). The role of adbuction in logic programming. *Handbook of Logic in Artificial Intelligence and Logic Programming*, 5:235–324.

[Kakas and Toni, 1999] Kakas, A. C. and Toni, F. (1999). Computing argumentation in logic programming. *Journal of Logic and Computation*, 9:515–562.

[Kowalski and Toni, 1996] Kowalski, R. A. and Toni, F. (1996). Abstract argumentation. *Artificial Intelligence and Law Journal*, 4(3–4):275–296.

[Labrou et al., 1999] Labrou, Y., Finin, T., and Peng, Y. (1999). Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45–52.

[Lakatos, 1970] Lakatos, I. (1970). Falsification and the methodology of scientific research programmes. In Lakatos, I. and Musgrave, A., editors, *Criticism and the Growth of Knowledge*, pages 91–195. Cambridge University Press.

[Lambert and Robertson, 2005] Lambert, D. and Robertson, D. (2005). Matchmaking and brokering multi-party interactions using historical performance data. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-agent Systems*.

[Levesque and Brachman, 1987] Levesque, H. J. and Brachman, R. J. (1987). Expressiveness and tractability in knowledge respresentation reasoning. *Computational Intelligence*, 3(2):78–93.

[Martin et al., 2010] Martin, P., Robertson, D., and Rovatsos, M. (2010). Opportunistic belief reconciliation during distributed interactions. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, pages 433–440.

[Maudet and Chaib-draa, 2002] Maudet, N. and Chaib-draa, B. (2002). Commitment-based and dialogue-game based protocols: New trends in agent communication languages. *The Knowledge Engineering Review*, 17(2).

[McDermott, 1982] McDermott, D. (1982). Nonmonotonic logic II: Nonmonotonic modal theories. *Journal of ACM*, 29:33–57.

[McGinnis et al., 2005] McGinnis, J., Robertson, D., and Walton, C. (2005). Protocol synthesis with dialogue structure theory. In *Proceedings of the 4th International Conference of Autonomous Agents and Multiagent Systems*, pages 1329–1330. ACM Press.

[Moore, 1985] Moore, R. C. (1985). Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25:75–94.

[O'Brien and Nicol, 1998] O'Brien, P. D. and Nicol, R. (1998). FIPA - towards a standard for software agents. *BT Technology Journal*, 16(3):51–59.

[Osman, 2003] Osman, N. (2003). Addressing constraint failures in distributed dialogue protocols. Master's thesis, School of Informatics, University of Edinburgh.

[Osman, 2007] Osman, N. (2007). Dynamic verification of trust in distributed open systems. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*.

[Patil et al., 1992] Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T., and Neches, R. (1992). The DARPA knowledge sharing effort: Progress report. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR '92)*.

[Pollock, 1995] Pollock, J. L. (1995). *Cognitive Carpentry*. The MIT Press.

[Prakken, 1995] Prakken, H. (1995). From logic to dialectics in legal argument. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law*, pages 165–174.

[Prakken, 2005] Prakken, H. (2005). Coherence and flexibility in dialogue games for argumentation. *Journal of Logic and Computation*, 15(6):1009–1040.

[Prakken and Sartor, 1995] Prakken, H. and Sartor, G. (1995). On the relation between legal language and legal argument: Assumptions, applicability and dynamic priorities. In *Proceedings of the 5th International Conference on Artificial Intelligence and Law*, pages 1–10.

[Rahwan and Larson, 2008] Rahwan, I. and Larson, K. (2008). Pareto optimality in abstract argumentation. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*.

[Rahwan et al., 2003] Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S., and Sonenberg, L. (2003). Argumentation-based negotiation. *The Knowledge Engineering Review*, 18(4):343–375.

[Reiter, 1980] Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 18:81–132.

[Robertson, 2004] Robertson, D. (2004). Multi-agent coordination as distributed logic programming. In *Proceedings of the 20th International Conference on Logic Programming*.

[Robertson et al., 2008] Robertson, D., Walton, C., Barker, A., Besana, P., Chen-Burger, Y. H., Hassan, F., Lambert, D., Li, G., McGinnis, J., Osman, N., Bundy, A., McNeill, F., van Harmelen, F., Sierra, C., and Giunchiglia, F. (2008). Models of interaction as a grounding for peer to peer knowledge sharing. *LNCS Advances in Web Semantics*, 1.

[Rosenschein and Zlotkin, 1994] Rosenschein, J. S. and Zlotkin, G. (1994). Designing conventions for automated negotiation. *AI Magazine*, 15(3):29–46.

[Russell and Norvig, 1995] Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice Hall.

[Searle, 1969] Searle, J. (1969). *Speech Acts: An Essay on the Philosophy of Language*. Cambridge University Press.

[Siebes et al., 2007] Siebes, R., Dupplaw, D., Kotoulas, S., de Pinnick, A. P., van Harmelen, F., and Robertson, D. (2007). The openknowledge system: An

interaction-centered approach to knowledge sharing. In *Proceedings of the 15th International Conferece on Cooperative Information Systems*.

[Simpson, 2006] Simpson, J., editor (2006). *Oxford English Dictionary*. Oxford University Press.

[Singh, 1998] Singh, M. P. (1998). Agent communication languages: Rethinking the principles. *Computer*, 21:40–47.

[Vreeswijk and Prakken, 2000] Vreeswijk, G. A. W. and Prakken, H. (2000). Credulous and sceptical argument games for preferred semantics. In *Proceedings of the 7th European Workshop on Logic for Artificial Intelligence*, pages 239–253.

[Walton, 2004a] Walton, C. (2004a). Multi-agent dialogue protocols. In *Proceedings of the 8th Interntional Symposium on Artificial Intelligence and Mathematics*.

[Walton, 2004b] Walton, C. D. (2004b). Model checking multi-agent web services. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services*.

[Walton and Krabbe, 1995] Walton, D. N. and Krabbe, E. C. W. (1995). *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press.

[Weiss, 2000] Weiss, G., editor (2000). *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. The MIT Press.

[Wooldridge, 2000] Wooldridge, M. (2000). Semnatic issues in the verification of agent communication languages. *Autonomous Agents and Multi-Agent Systems*, 3(1):9–31.