Opportunistic Argumentation During Distributed Multi-Agent Interaction

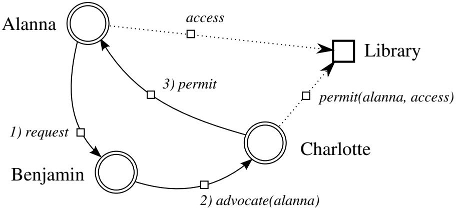
Paul Martin David Robertson Michael Rovatsos

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

CISA Seminar, 14th of June 2010

In Abstract

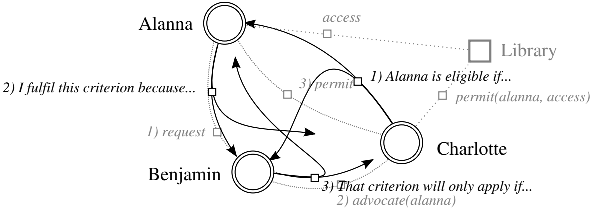
Dialogue protocols can be used to coordinate agent interaction:



The outcome of an interaction is determined by the constraints imposed by its protocol.

In Abstract

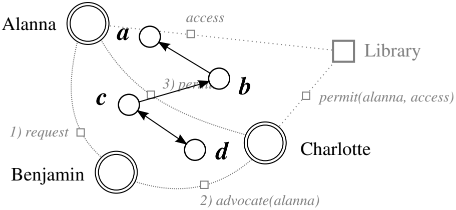
Discursive dialogue can be used to augment protocols in situ:



Argumentation can drive the discussion of constraints on interaction.

In Abstract

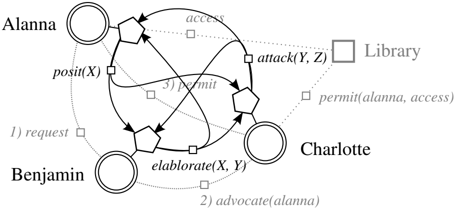
In complex domains however, it is necessary to restrict dialogue:



A sufficiently expressive argument space permits the synchronisation of agent beliefs within that space.

In Abstract

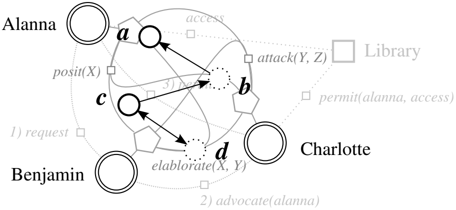
A distributed mechanism for generating such dialogue during interactions can be produced:



The argument space for an interaction portrayal is refined over the course of interaction.

In Abstract

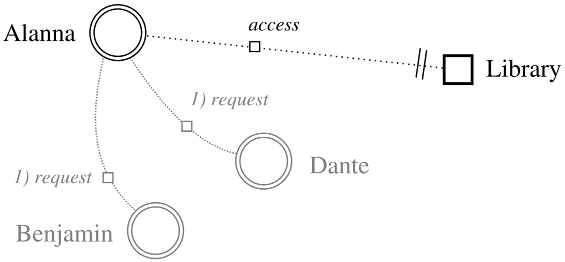
In essense, we have a system for distributed belief revision on demand:



The interaction portrayal is merely a medium for concept dissemination and truth maintenance.

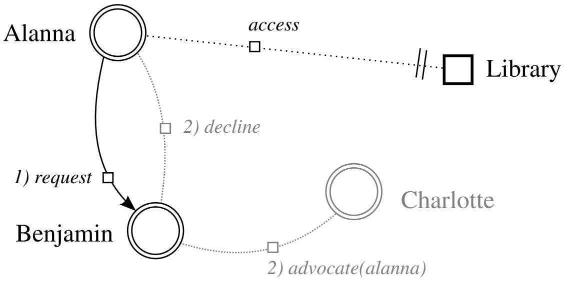
A Simple Interaction . . .

Alanna desires access to a resource . . .



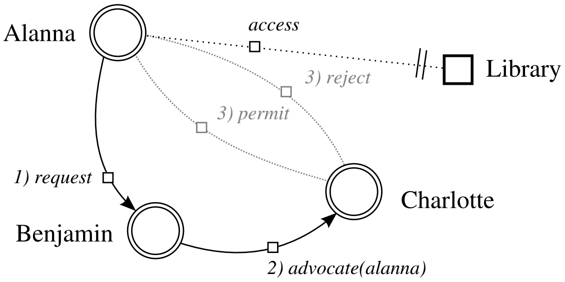
A Simple Interaction . . .

Alanna enlists Benjamin's assistance . . .



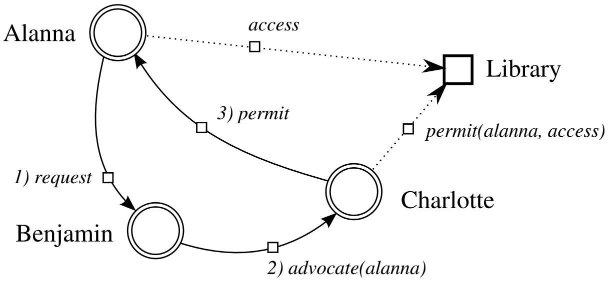
A Simple Interaction . . .

Benjamin recommends Alanna to Charlotte . . .



A Simple Interaction . . .

Charlotte then grants access to Alanna.



A Simple Interaction . . . ?

The outcome of any interaction however, is dependent on the environment and on agents understanding the intentions of their peers.

- Agents must be able to model possible interaction outcomes.
- The responses of agents must conform to those models.
- The use of shared protocols and social norms make it easier to map responses to outcomes.

There exist protocol languages (such as *LCC*) and protocol distribution mechanisms (such as within *OpenKnowledge*) to facilitate this.

A Simple Interaction Protocol

The previous interaction can be modelled by the following LCC protocol:

$$
\begin{array}{l}
\mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) :: \\
\quad \mathrm{request} \Rightarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) \\
\quad \leftarrow \neg\ \mathrm{accessible}(\mathit{Applicant},\ \mathit{Resource}) \wedge \mathrm{patron}(\mathit{Advocate},\ \mathit{Resource})\ \mathrm{then} \\
\quad (\quad \mathrm{fail} \leftarrow \mathrm{decline} \Leftarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) \vee \\
\qquad\qquad\quad \mathrm{reject} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \\
\qquad \mathrm{else} \\
\qquad \mathrm{succeed} \leftarrow \mathrm{permit} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \wedge \\
\qquad\qquad\qquad \mathrm{accessible}(\mathit{Applicant},\ \mathit{Resource}) \quad).
\end{array}
$$

$$
\begin{array}{l}
\mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) :: \\
\quad \mathrm{request} \Leftarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant})\ \mathrm{then} \\
\quad (\quad \mathrm{advocate}(\mathit{Applicant}) \Rightarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \\
\qquad \leftarrow \mathrm{controller}(\mathit{Controller},\ \mathit{Resource}) \wedge \mathrm{trustworthy}(\mathit{Applicant},\ \mathrm{access}(\mathit{Resource})) \\
\qquad \mathrm{else} \\
\qquad \mathrm{decline} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad).
\end{array}
$$

$$
\begin{array}{l}
\mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) :: \\
\quad \mathrm{advocate}(\mathit{Applicant}) \Leftarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate})\ \mathrm{then} \\
\quad (\quad \mathrm{permit}(\mathit{Applicant},\ \mathrm{access}(\mathit{Resource})) \\
\qquad \leftarrow \mathrm{controls}(\mathit{Controller},\ \mathrm{permissions}(\mathit{Resource})) \wedge \\
\qquad\quad \mathrm{trusts}(\mathit{Controller},\ \mathit{Advocate}) \wedge \mathrm{eligible}(\mathit{Applicant},\ \mathrm{access}(\mathit{Resource}))\ \mathrm{then} \\
\qquad \mathrm{permit} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad) \\
\qquad \mathrm{else} \\
\qquad \mathrm{reject} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad).
\end{array}
$$

A Simple Interaction Protocol

In this protocol, the outcome of interaction is determined by the satisfiability of certain logical propositions:

$$
\begin{array}{l}
\mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) :: \\
\quad \mathrm{request} \Rightarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) \\
\quad \leftarrow \neg\ \textbf{accessible(Applicant, Resource)} \wedge \textbf{patron(Advocate, Resource)}\ \mathrm{then} \\
\quad (\quad \mathrm{fail} \leftarrow \mathrm{decline} \Leftarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) \vee \\
\qquad\qquad\quad \mathrm{reject} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \\
\qquad \mathrm{else} \\
\qquad \mathrm{succeed} \leftarrow \mathrm{permit} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \wedge \\
\qquad\qquad\qquad \mathrm{accessible}(\mathit{Applicant},\ \mathit{Resource}) \quad).
\end{array}
$$

$$
\begin{array}{l}
\mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate}) :: \\
\quad \mathrm{request} \Leftarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant})\ \mathrm{then} \\
\quad (\quad \mathrm{advocate}(\mathit{Applicant}) \Rightarrow \mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) \\
\qquad \leftarrow \textbf{controller(Controller, Resource)} \wedge \textbf{trustworthy(Applicant, access(Resource))} \\
\qquad \mathrm{else} \\
\qquad \mathrm{decline} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad).
\end{array}
$$

$$
\begin{array}{l}
\mathrm{a}(\mathrm{controller}(\mathit{Resource}),\ \mathit{Controller}) :: \\
\quad \mathrm{advocate}(\mathit{Applicant}) \Leftarrow \mathrm{a}(\mathrm{advocate}(\mathit{Resource}),\ \mathit{Advocate})\ \mathrm{then} \\
\quad (\quad \mathrm{permit}(\mathit{Applicant},\ \mathrm{access}(\mathit{Resource})) \\
\qquad \leftarrow \textbf{controls(Controller, permissions(Resource))} \wedge \\
\qquad\quad \textbf{trusts(Controller, Advocate)} \wedge \textbf{eligible(Applicant, access(Resource))}\ \mathrm{then} \\
\qquad \mathrm{permit} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad) \\
\qquad \mathrm{else} \\
\qquad \mathrm{reject} \Rightarrow \mathrm{a}(\mathrm{applicant}(\mathit{Resource}),\ \mathit{Applicant}) \quad).
\end{array}
$$

A Simple Interaction Protocol

In a complex environment, agents may evaluate the same proposition differently based on the information accessible to them and any abductions they might make:

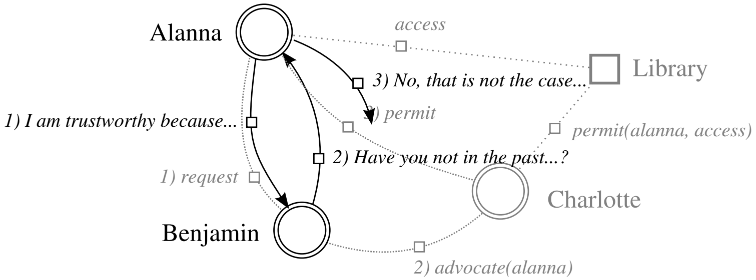
a(advocate(*Resource*), *Advocate*) ::
request $\Leftarrow$ a(applicant(*Resource*), *Applicant*) then
(advocate(*Applicant*) $\Rightarrow$ a(controller(*Resource*), *Controller*)
$\leftarrow$ controller(*Controller*, *Resource*) $\wedge$ **trustworthy(Applicant, access(Resource))**
else
decline $\Rightarrow$ a(applicant(*Resource*), *Applicant*)).

trustworthy(alanna, access(library)) — *because Alanna is an accredited researcher and there is no benefit to Alanna abusing the trust placed on her.*

¬trustworthy(alanna, access(library)) — *because Alanna has abused trust in similar circumstances in the past.*

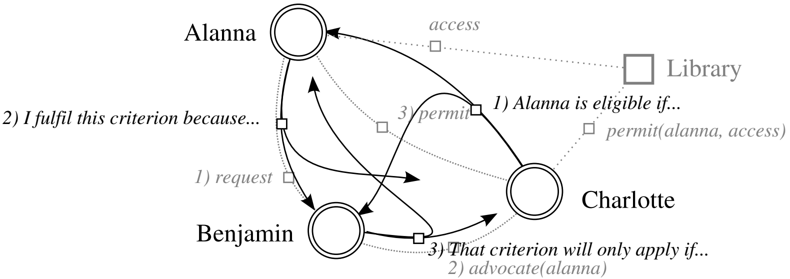
Interaction is Not Simple

Would like to be able to discuss constraints on interaction *during* interaction.



Strict adherence to a protocol does not permit free-form discussion however.

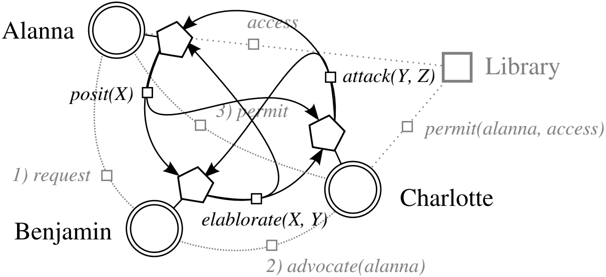
Discussion takes many forms (persuasion, inquiry, dissemination, *etc.*).



Encoding such discussion into a protocol is too confining.

Interaction is Not Simple

Instead, it would be preferable to permit domain-specific discussion alongside generic protocols.



We can do this by composing a decision problem, the solving of which produces the desired dialogue.

Interaction is Not Simple

This decision problem can be seen as one of disputation.

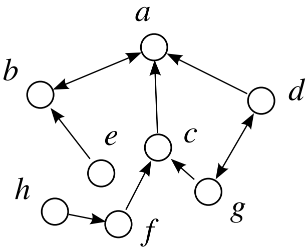
- Positing of expectations invites dispute.
- Dispute forces elaboration upon reasoning.
- Elaboration propagates new premises and rules.

Disputation can be driven by a process of argumentation.

Abstract Argumentation

Argumentation is concerned with the selection of jointly consistent hypotheses from a hypothesis space.

- A system of arguments is produced with an argumentation framework.

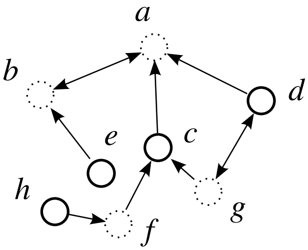


A **system of arguments** $(\mathcal{A}, \rightharpoonup)$ *consists of a set of arguments* $\mathcal{A}$ *and a set of attacks* $\mathbf{a} \rightharpoonup \mathbf{b}$, *where* $\mathbf{a}, \mathbf{b} \in \mathcal{A}$.

Abstract Argumentation

Argumentation is concerned with the selection of jointly consistent hypotheses from a hypothesis space.

- The argument system is then interpreted and arguments labelled.

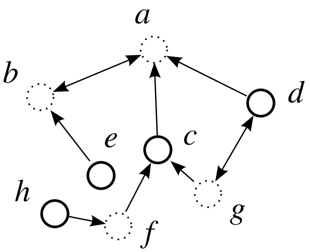


An **extension** $\mathcal{E} \subseteq \mathcal{A}$ *of an argument system* $(\mathcal{A}, \rightharpoondown)$ *is any subset of arguments considered to be collectively acceptable according to some chosen* **acceptability semantics**.

Abstract Argumentation

Argumentation is concerned with the selection of jointly consistent hypotheses from a hypothesis space.

- A defeasible theory is then inferred from accepted arguments.



A **theory** Π *will be constructed from the hypotheses used in arguments in an accepted extension* $\mathcal{E}$*, based on the internal structure of those arguments.*

Argumentation Frameworks

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

- $(\mathcal{L}, \vdash)$ is a logical framework used to construct arguments.
- Δ is the **argument space** within which arguments are generated.

$(\mathcal{L}, \vdash, \Delta)$ can be used to generate a system of arguments $(\mathcal{A}, \rightharpoondown)$ such that for every argument $\mathbf{a} \in \mathcal{A}$, it is the case that $\mathbf{a} \in \Delta$.

Assumption-Based Argumentation

An **assumption-based** argumentation framework can be described by a tuple $(\mathcal{L}, \vdash, \Delta)$ where:

- $(\mathcal{L}, \vdash)$ is a deductive framework ($\vdash$ is monotonic).
- An argument is a pair $\langle \Phi, \alpha \rangle \in \Delta$ where $\Phi \subseteq \mathcal{L}$, there exists no sentence $\varphi \in \Phi$ such that $\Phi \vdash (\varphi \wedge \neg\varphi)$, it is the case that $\Phi \vdash \alpha$ and there exists no subset $\Phi' \subset \Phi$ such that $\Phi' \vdash \alpha$.
- $\langle \Phi, \alpha \rangle \rightharpoondown \langle \Psi, \beta \rangle$ if and only if $\Psi \vdash \gamma$ and $\{\alpha\} \vdash \neg\gamma$ for some sentence $\gamma \in \mathcal{L}$.
- $\langle \Phi, \alpha \rangle \in \Delta$ if $\Phi \subseteq H$, for some set of assumptions H defined by Δ.

We can use assumption-based argumentation to describe the formulation of an agent's beliefs.

Assumption-Based Argumentation

A deductive framework $(\mathcal{L}, \vdash)$ can be augmented with domain-specific axioms and rules:

$$
\begin{array}{l}
\mathrm{agent}(\mathrm{alanna}) \\
\mathrm{resource}(\mathrm{library}) \\
\ldots \\
\forall X, Y.\ \mathrm{patron}(X, Y) \rightarrow \mathrm{agent}(X) \wedge \mathrm{resource}(Y) \\
\forall X, Y.\ \mathrm{controller}(X, Y) \rightarrow \mathrm{controls}(X, \mathrm{permissions}(Y)) \\
\forall X, Y.\ \mathrm{assignable}(X, Y) \rightarrow \exists Z.\ \mathrm{component}(Z, Y) \wedge \neg \mathrm{broken}(Z) \\
\ldots
\end{array}
$$

An argument space Δ can then be produced by generating a set of base hypotheses H:

$$
\begin{array}{l}
\text{controller}(\text{charlotte}, \text{library}) \\
\text{domain}(\text{library}, \text{informatics}) \\
\text{uses}(\text{benjamin}, \text{library}) \\
\ldots \\
\forall X, Y.\ \text{user}(X, Y) \rightarrow \text{patron}(X, Y) \\
\forall X, Y, Z.\ \text{researcher}(X, Y) \wedge \text{domain}(Z, Y) \rightarrow \text{eligible}(X, \text{access}(Z)) \\
\forall X, Y.\ \text{beneficial}(X, \text{abuse}(Y)) \wedge \text{trustworthy}(X, Y) \rightarrow \text{false} \\
\ldots
\end{array}
$$

These hypotheses might be based on historical observation, induction, reasoning by analogy, social discourse, *etc.*

Using the resulting argumentation framework $(\mathcal{L}, \vdash, \Delta)$, a system of arguments $(\mathcal{A}, \rightharpoondown)$ can then be generated which describes the environment:

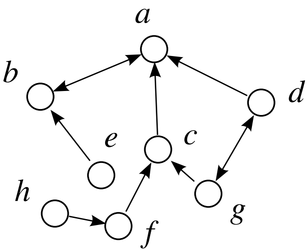
$$\begin{aligned}
\mathbf{a} = \langle \ \{ \ & \mathit{needs}(\mathit{alanna}, \mathit{access}(\mathit{library})), \\
& \mathit{trusts}(\mathit{eliza}, \mathit{alanna}), \\
& \ldots \\
& \forall X, Y.\ \mathit{beneficial}(X, \mathit{abuse}(Y)) \land \mathit{trustworthy}(X, Y) \rightarrow \mathit{false} \ \}, \\
\mathit{trustworthy}&(\mathit{alanna}, \mathit{access}(\mathit{library})) \ \rangle
\end{aligned}$$

$$\begin{aligned}
\mathbf{b} = \langle \ \{ \ & \mathit{requirement}(\mathit{alanna}, \mathit{data}), \\
& \mathit{source}(\mathit{data}, \mathit{library}), \\
& \ldots \\
& \forall X, Y.\ \mathit{requirement}(X, Y) \land \exists! Z.\ \mathit{source}(Y, Z) \rightarrow \mathit{needs}(X, Z) \ \}, \\
\neg \mathit{needs}&(\mathit{alanna}, \mathit{access}(\mathit{library})) \ \rangle
\end{aligned}$$

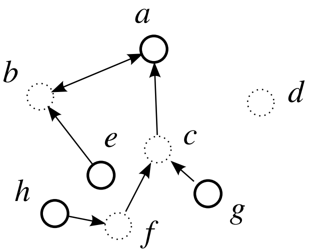
...

Assumption-Based Argumentation

The set of observations Θ of the environment acts as a hard core around which theory Π is constructed; when interpreting the argument system, arguments which contradict Θ are automatically rejected.



The set of observations Θ of the environment acts as a hard core around which theory Π is constructed; when interpreting the argument system, arguments which contradict Θ are automatically rejected.



Assumption-Based Argumentation

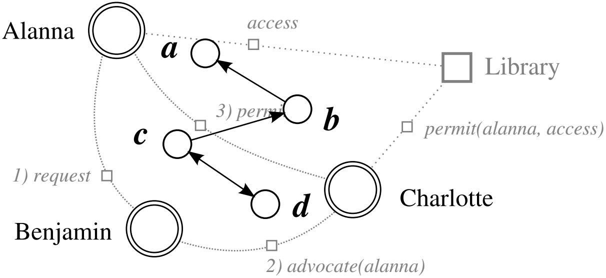
The **context** $\mathcal{C}$ for a theory Π can be described as a persistent argumentation process by a tuple $(\Theta, (\mathcal{L}, \vdash, \Delta), (\mathcal{A}, \rightharpoondown), \mathit{lab})$:

- Θ is the core of Π.
- $(\mathcal{L}, \vdash, \Delta)$ is an assumption-based argumentation framework in which $(\mathcal{A}, \rightharpoondown)$ is generated.
- $(\mathcal{A}, \rightharpoondown)$ is a system of arguments, an interpretation of which describes Π.
- An argument labelling function $\mathit{lab}(\Theta) : \mathcal{A} \rightharpoonup \{\text{in}, \text{out}\}$ is a partial function which interprets $(\mathcal{A}, \rightharpoondown)$ according to chosen acceptability semantics given prior acceptance of Θ.

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

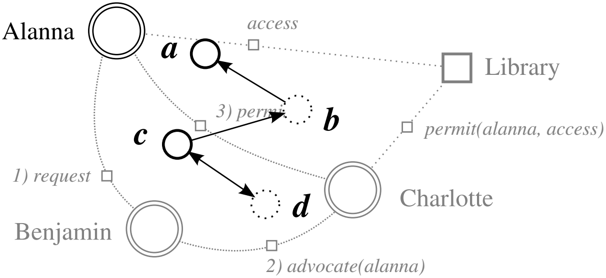
Distributed Argumentation

Ideally, agents should be able to generate a shared system of arguments within an argumentation framework defined by an ongoing interaction.



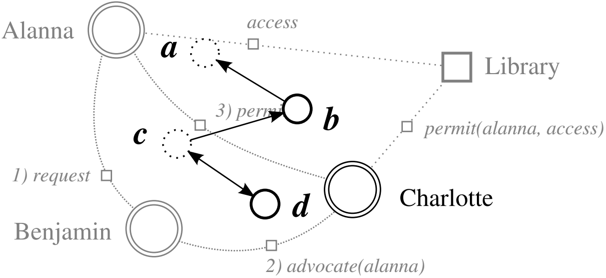
There are two notable problems however.

The first problem is that there may still be multiple admissible interpretations of the shared argument system even after all the evidence is compiled.



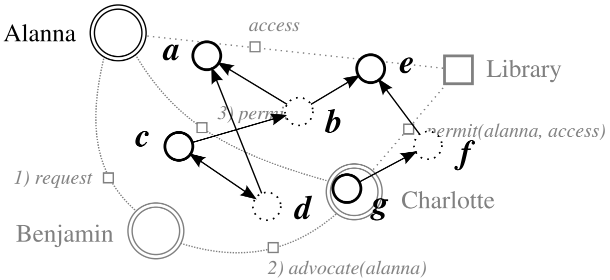
Alanna chooses to accept one thing . . .

The first problem is that there may still be multiple admissible interpretations of the shared argument system even after all the evidence is compiled.



. . . Charlotte chooses to accept another.

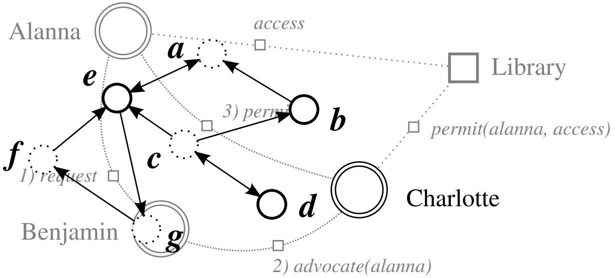
The second problem is that the argument spaces in which agents individually reason may be difficult to reconcile within a single space.



It may be computationally prohibitive to reconcile all of Alanna's assumptions . . .

Distributed Argumentation

The second problem is that the argument spaces in which agents individually reason may be difficult to reconcile within a single space.



...with all of Charlotte's.

Distributed Argumentation

Different agents deconstruct their views of the world differently.

$$
\begin{aligned}
\mathbf{a}_1 = \langle\ \{\ & \mathit{employed}(\mathit{alanna}, \mathit{edinburgh}),\\
& \mathit{publication}(\mathit{alanna}, \mathit{paper3}, \mathit{aamas11}),\\
& \ldots\\
& \forall X, Y.\mathit{published}(X, Y) \wedge \exists Z.\ \mathit{tenure}(X, Z) \rightarrow \mathit{researcher}(X, Y),\\
& \ldots\\
& \forall X, Y, Z.\ \mathit{researcher}(X, Y) \wedge \mathit{domain}(Z, Y) \rightarrow \mathit{eligible}(X, \mathit{access}(Z))\ \},\\
& \mathit{eligible}(\mathit{alanna}, \mathit{access}(\mathit{library}))\ \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{a}_2 = \langle\ \{\ & \mathit{graduate}(\mathit{alanna}, \mathit{mars}),\\
& \mathit{project\_time}(\mathit{projectX}, \mathit{date}(14, 6, 2010)),\\
& \ldots\\
& \forall X, Y.\mathit{expert}(X, Y) \wedge \mathit{active\_in\_field}(X, Y) \rightarrow \mathit{researcher}(X, Y),\\
& \ldots\\
& \forall X, Y, Z.\ \mathit{researcher}(X, Y) \wedge \mathit{domain}(Z, Y) \rightarrow \mathit{eligible}(X, \mathit{access}(Z))\ \},\\
& \mathit{eligible}(\mathit{alanna}, \mathit{access}(\mathit{library}))\ \rangle
\end{aligned}
$$

The underlying reasoning behind otherwise identical claims may be drastically different.

However the basic reason for accepting a given claim might be the same regardless.

$$
\begin{aligned}
\mathbf{a}_1 = \langle\, \{\ & \mathit{employed}(\mathit{alanna}, \mathit{edinburgh}),\\
& \mathit{publication}(\mathit{alanna}, \mathit{paper3}, \mathit{aamas11}),\\
& \ldots\\
& \forall X, Y.\mathit{published}(X, Y) \wedge \exists Z.\ \mathit{tenure}(X, Z) \rightarrow \mathbf{researcher}(\mathbf{X}, \mathbf{Y}),\\
& \ldots\\
& \forall \mathbf{X}, \mathbf{Y}, \mathbf{Z}.\ \mathbf{researcher}(\mathbf{X}, \mathbf{Y}) \wedge \mathbf{domain}(\mathbf{Z}, \mathbf{Y}) \rightarrow \mathbf{eligible}(\mathbf{X}, \mathbf{access}(\mathbf{Z}))\ \},\\
& \mathbf{eligible}(\mathbf{alanna}, \mathbf{access}(\mathbf{library}))\ \rangle
\end{aligned}
$$

$$
\begin{aligned}
\mathbf{a}_2 = \langle\, \{\ & \mathit{graduate}(\mathit{alanna}, \mathit{mars}),\\
& \mathit{project\_time}(\mathit{projectX}, \mathit{date}(14, 6, 2010)),\\
& \ldots\\
& \forall X, Y.\mathit{expert}(X, Y) \wedge \mathit{active\_in\_field}(X, Y) \rightarrow \mathbf{researcher}(\mathbf{X}, \mathbf{Y}),\\
& \ldots\\
& \forall \mathbf{X}, \mathbf{Y}, \mathbf{Z}.\ \mathbf{researcher}(\mathbf{X}, \mathbf{Y}) \wedge \mathbf{domain}(\mathbf{Z}, \mathbf{Y}) \rightarrow \mathbf{eligible}(\mathbf{X}, \mathbf{access}(\mathbf{Z}))\ \},\\
& \mathbf{eligible}(\mathbf{alanna}, \mathbf{access}(\mathbf{library}))\ \rangle
\end{aligned}
$$

In essense, we don't want to waste time on details if agents are in broad agreement about how to proceed with interaction.

Potential Argumentation

Given a logical framework $(\mathcal{L}, \vdash)$, an argument **a** is considered to be a **potential argument** in relation to another argument **b** (*i.e.* $\mathbf{a} \sqsubseteq \mathbf{b}$) if:

- For every attack $\mathbf{c} \rightharpoondown \mathbf{a}$, it is the case that $\mathbf{c} \rightharpoondown \mathbf{b}$.
- If $\mathbf{a} \rightharpoondown \mathbf{d}$, then $\mathbf{b} \rightharpoondown \mathbf{d}$ and *vice versa*.

Basically **a** is an argument which could be elaborated upon to produce **b**.

Potential Argumentation

Consider the following argument:

$$\begin{aligned}\mathbf{a}' = \langle\ \{\ & \mathit{researcher}(\mathit{alanna}, \mathit{inforamtics}),\\ & \mathit{domain}(\mathit{library}, \mathit{informatics})\\ & \forall X, Y, Z.\ \mathit{researcher}(X, Y) \wedge \mathit{domain}(Z, Y) \rightarrow \mathit{eligible}(X, \mathit{access}(Z))\ \},\\ & \mathit{eligible}(\mathit{alanna}, \mathit{access}(\mathit{library}))\ \rangle\end{aligned}$$

Given arguments $\mathbf{a}_1$ and $\mathbf{a}_2$ earlier, $\mathbf{a}' \sqsubset \mathbf{a}_1$ and $\mathbf{a}' \sqsubset \mathbf{a}_2$.

Potential Argumentation

Given a logical framework $(\mathcal{L}, \vdash)$, a **potential restriction** of a set of arguments S into an argument space Δ 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 $\mathbf{c} \sqsubseteq \mathbf{b}$.

Thus given a restricted argument space Δ, we can prune irrelevant arguments and simplify those remaining as demanded by Δ.

Belief Synchronisation

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 theory context $\mathcal{C}_i$ such that Π_i is derived from an admissible extension $\mathcal{E}_i$ of $\mathcal{C}$.
- 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 $\mathcal{C}_j$.

Given that we cannot impose one interpretation on all agents, synchronisation provides an alternative 'goal state' for dialogue.

Sufficient Expressivity

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 context $\mathcal{C}$ such that Π is derived from an admissible extension $\mathcal{E}$ of $\mathcal{C}$.
- There exists a potential restriction $\mathcal{E}'$ of $\mathcal{E}$ into Δ.
- If there exists an argument $\mathbf{a} \in \Delta$ such that $\mathbf{a} \rightharpoondown \mathbf{b}$ for some argument $\mathbf{b} \in \mathcal{E}'$, then provided that there exists an argument $\mathbf{c} \in \mathcal{E}$ such that $\mathbf{c} \rightharpoondown \mathbf{a}$, there exists an argument $\mathbf{d} \in \Delta$ such that $\mathbf{d} \sqsubseteq \mathbf{c}$ and $\mathbf{d} \rightharpoondown \mathbf{a}$.

If an argument space Δ is sufficiently expressive for all theories held by agents engaged in interaction, then it can be proven that those theories can be synchronised within Δ.

Argument Reconciliation

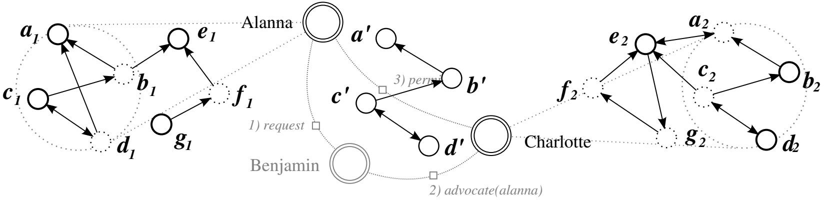
A system of arguments $(\mathcal{A}, \rightharpoondown)$ within an argument space Δ is **reconciled** with a theory context $\mathcal{C}$ if and only if:

- Every complete extension of $(\mathcal{A}, \rightharpoondown)$ 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 $(\mathcal{A}, \rightharpoondown)$ given Θ.

If $(\mathcal{A}, \rightharpoondown)$ can be reconciled with the context of every agent engaged in interaction, then it can be proven that every theory Π derived from a context $\mathcal{C}$ is synchronised within Δ.

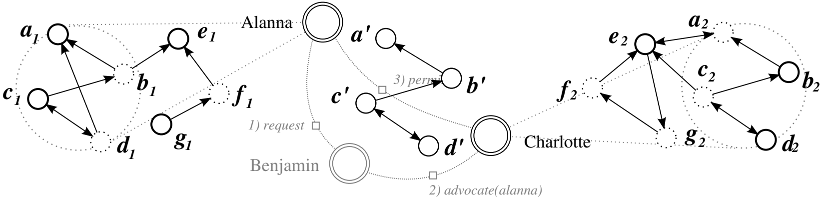
Interaction Portrayals

In theory, we should be able to map arguments from many separate theory contexts into a single restricted argument space and then map the potential restrictions back into those separate spaces.



If the argument space is sufficiently expressive, then it should be possible to then synchronise the theories produced by those theory contexts within the restricted argument space.

In theory, we should be able to map arguments from many separate theory contexts into a single restricted argument space and then map the potential restrictions back into those separate spaces.



The most immediate concern is how to determine the argument space in practice.

Interaction Portrayals

We can focus the argument space Δ of a shared argument system initially on portrayable propositions in the protocol for an interaction:

$$
\begin{array}{l}
\mathrm{a}(\mathrm{applicant}(\mathrm{library}),\ \mathrm{alanna}) :: \\
\quad \mathrm{request} \Rightarrow \mathrm{a}(\mathrm{advocate}(\mathrm{library}),\ Advocate) \\
\quad \leftarrow \neg\ \mathbf{accessible(alanna,\ library)} \wedge \mathbf{patron(Advocate,\ library)}\ \mathrm{then} \\
\quad (\quad \mathrm{fail} \leftarrow \mathrm{decline} \Leftarrow \mathrm{a}(\mathrm{advocate}(\mathrm{library}),\ Advocate) \vee \\
\qquad\qquad \mathrm{reject} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathrm{library}),\ Controller) \\
\qquad \mathrm{else} \\
\qquad \mathrm{succeed} \leftarrow \mathrm{permit} \Leftarrow \mathrm{a}(\mathrm{controller}(\mathrm{library}),\ Controller) \wedge \\
\qquad\qquad\qquad \mathrm{accessible}(\mathrm{alanna},\ \mathrm{library}) \quad).
\end{array}
$$

$$
\begin{array}{l}
\mathrm{a}(\mathrm{advocate}(Resource),\ Advocate) :: \\
\quad \mathrm{request} \Leftarrow \mathrm{a}(\mathrm{applicant}(Resource),\ Applicant)\ \mathrm{then} \\
\quad (\quad \mathrm{advocate}(Applicant) \Rightarrow \mathrm{a}(\mathrm{controller}(Resource),\ Controller) \\
\qquad \leftarrow \mathrm{controller}(Controller,\ Resource) \wedge \mathrm{trustworthy}(Applicant,\ \mathrm{access}(Resource)) \\
\qquad \mathrm{else} \\
\qquad \mathrm{decline} \Rightarrow \mathrm{a}(\mathrm{applicant}(Resource),\ Applicant) \quad).
\end{array}
$$

Interaction Portrayals

As interaction proceeds, Δ can be expanded in breadth to encompass any additional propositions as they become portrayable:

a(applicant(library), alanna) ::
  request $\Rightarrow$ a(advocate(library), benjamin)
  $\leftarrow \neg$ **accessible(alanna, library)** $\wedge$ **patron(benjamin, library)** then
  (fail $\leftarrow$ decline $\Leftarrow$ a(advocate(library), benjamin) $\vee$
      reject $\Leftarrow$ a(controller(library), *Controller*)
    else
    succeed $\leftarrow$ permit $\Leftarrow$ a(controller(library), *Controller*) $\wedge$
      accessible(alanna, library)).

a(advocate(library), benjamin) ::
  request $\Leftarrow$ a(applicant(library), alanna) then
  (advocate(alanna) $\Rightarrow$ a(controller(library), *Controller*)
    $\leftarrow$ **controller(Controller, library)** $\wedge$ **trustworthy(alanna, access(library))**
    else
    decline $\Rightarrow$ a(applicant(library), alanna)).

Initially, we might only permit arguments of highest potential within Δ:

$\ldots$

$\mathbf{b} = \langle\ \{\ \mathit{patron}(\mathit{benjamin}, \mathit{library})\ \},\ \mathit{patron}(\mathit{benjamin}, \mathit{library})\ \rangle$

$\mathbf{c} = \langle\ \{\ \mathit{patron}(\mathit{dante}, \mathit{library})\ \},\ \mathit{patron}(\mathit{dante}, \mathit{library})\ \rangle$

$\ldots$

$$\begin{aligned}\mathbf{e} = \langle\ \{\ & \mathit{trustworthy}(\mathit{alanna}, \mathit{access}(\mathit{library}))\ \},\\ & \mathit{trustworthy}(\mathit{alanna}, \mathit{access}(\mathit{library}))\ \rangle\end{aligned}$$

Interaction Portrayals

However potential arguments can be elaborated upon as necessary to permit attack and to defend from attack, thus expanding Δ in depth:

$$\ldots$$

$$\begin{aligned}\mathbf{e} = \langle\ \{\ & \neg beneficial(alanna, abuse(access(library))),\\ & \forall X, Y.\ beneficial(X, abuse(Y)) \wedge trustworthy(X, Y) \rightarrow false\ \},\\ & trustworthy(alanna, access(library))\ \rangle\end{aligned}$$

$$\begin{aligned}\mathbf{f} = \langle\ \{\ & beneficial(alanna, abuse(access(library)))\ \},\\ & beneficial(alanna, abuse(access(library)))\ \rangle\end{aligned}$$

However potential arguments can be elaborated upon as necessary to permit attack and to defend from attack, thus expanding Δ in depth:

$$\ldots$$

$$\begin{aligned}
\mathbf{e} = \langle\ \{\ & \neg beneficial(alanna, abuse(access(library))), \\
& \forall X, Y.\ beneficial(X, abuse(Y)) \wedge trustworthy(X, Y) \rightarrow false\ \}, \\
& trustworthy(alanna, access(library))\ \rangle
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{g} = \langle\ \{\ & source(laboratory, data), \\
& sellable(benjamin, dante, data), \\
& \neg beneficial(benjamin, abuse(access(laboratory)))\ \}, \\
& source(laboratory, data) \wedge sellable(benjamin, dante, data) \wedge \\
& \neg beneficial(benjamin, abuse(access(laboratory)))\ \rangle
\end{aligned}$$

Interaction Portrayals

An interaction portrayal $\mathcal{P}$ is an (annotated) shared system of arguments $(\mathcal{A}, \rightharpoonup)$ used to describe the constraints upon an interaction $\mathcal{I}$.

- As interaction progresses, agents add arguments into $\mathcal{P}$ if those argument are within $\mathcal{P}$'s argument space Δ.
  - If peers can make counter-arguments, they may do so, being now within Δ.
  - If an added argument is unknown to a peer, then that agent should expand its context so as to account for it, possibly updating its beliefs.

We need only ensure that the argument space is able to grow to encompass relevant arguments without becoming too permissive.

Interaction Portrayals

An interaction portrayal $\mathcal{P}$ is an (annotated) shared system of arguments $(\mathcal{A}, \rightharpoondown)$ used to describe the constraints upon an interaction $\mathcal{I}$.

- Prior to resolution of a constraint on $\mathcal{I}$ by a given agent, $\mathcal{P}$ must be reconciled with the contexts of all agents involved in $\mathcal{I}$.
  - Upon reconciliation, the given agent must resolve any constraint to its own satisfaction.
  - This may introduce new portrayable propositions, expanding Δ.
- If the environment state changes, this may generate new arguments.

At the end of interaction, $\mathcal{P}$ can be discarded, its information having been assimilated by all peers in $\mathcal{I}$.

Interaction Portrayals

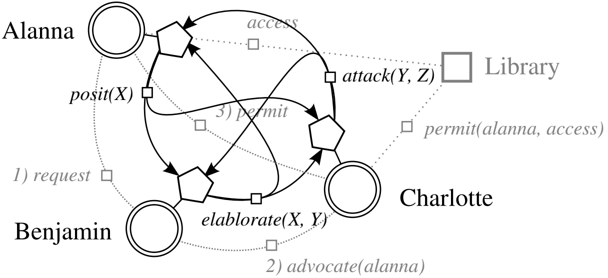
An argument **a** is within the argument space Δ of a portrayal $\mathcal{P}$ with respect to a context $\mathcal{C}$ (*i.e.* $\mathbf{a} \in \Delta$ from the perspective of an agent σ in context $\mathcal{C}$) if and only if:

- α is the claim of **a**, and either α or $\neg\alpha$ is portrayable; or there exists an argument $\mathbf{b}' \in \mathcal{P}$ such that $\mathbf{a} \rightharpoondown \mathbf{b}$, where $\mathbf{b} \in \mathcal{C}$ and $\mathbf{b}' \sqsubseteq \mathbf{b}$.
- There does not exist an elaboration $\mathbf{c} \in \mathcal{E}_A$, where $\mathcal{E}_A$ is the accepted extension of $\mathcal{C}$, such that $\mathbf{a} \sqsubseteq \mathbf{c}$ and $\mathbf{d} \rightharpoondown \mathbf{c}$ for some argument $\mathbf{d} \in \mathcal{P}$, but $\mathbf{d} \not\rightharpoondown \mathbf{a}$, *unless* there also exists an elaboration $\mathbf{e} \in \mathcal{E}$ such that $\mathbf{a} \sqsubset \mathbf{e} \not\sqsubseteq \mathbf{c}$ and $\mathbf{d} \not\rightharpoondown \mathbf{e}$.
- There does not exist a potential argument **f** such that: $\mathbf{f} \sqsubset \mathbf{a}$, but $\mathbf{f} \not\sqsubseteq \mathbf{g}$ for some argument $\mathbf{g} \in \mathcal{P}$; and $\mathbf{f} \in \Delta$.

It can be proven that Δ will always become sufficiently expressive with respect to the theory Π derived from $\mathcal{C}$ as arguments are posited into $\mathcal{P}$.

A Distributed Portrayal Mechanism

We can implement interaction portrayals in a distributed system by distributing instances of a portrayal to all peers in an interaction.



The (asynchronous) messages used to keep every instance updated constitute the dialogues we have been interested in from the beginning.

A Distributed Portrayal Mechanism

A **portrayal instance** $\mathcal{P}[\sigma]$ used by an agent σ in context $\mathcal{C}$ to describe a portrayal $\mathcal{P}$ of an interaction $\mathcal{I}$ is described by a tuple $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoondown), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$ 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.
- $(\mathcal{A}, \rightharpoondown)$ is the system of arguments described by $\mathcal{P}$ such that:
  - An argument is a pair $\langle \Phi, \alpha \rangle \in \mathcal{A}$ wherein $\Phi \subseteq \mathcal{L}$ and $\Phi \vdash \alpha$ according to $\mathcal{C}_\mu$ of an agent $\mu \in \Sigma$.
  - An attack is a relation $\langle \Phi, \alpha \rangle \rightharpoondown \langle \Psi, \beta \rangle$ between two arguments $\langle \Phi, \alpha \rangle$, $\langle \Psi, \beta \rangle \in \mathcal{A}$ such that $\Psi \vdash \gamma$ and $\{\alpha\} \vdash \neg\gamma$ for some sentence $\gamma \in \mathcal{L}$ according to $\mathcal{C}_\mu$ of an agent $\mu \in \Sigma$.

A Distributed Portrayal Mechanism

A **portrayal instance** $\mathcal{P}[\sigma]$ used by an agent σ in context $\mathcal{C}$ to describe a portrayal $\mathcal{P}$ of an interaction $\mathcal{I}$ is described by a tuple $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoondown), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$ where:

- The **invalid argument** function $\mathsf{inv}_{\mathcal{A}} : \Sigma \rightarrow 2^{\mathcal{A}}$ 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 $\mathcal{C}_{\mu}$.

This function is used to identify where agents are presuming certain axioms or domain rules not recognised by their peers.

A Distributed Portrayal Mechanism

A **portrayal instance** $\mathcal{P}[\sigma]$ used by an agent σ in context $\mathcal{C}$ to describe a portrayal $\mathcal{P}$ of an interaction $\mathcal{I}$ is described by a tuple $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoondown), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$ where:

- The **invalid attack** function $\mathsf{inv}_{\rightharpoondown} : \Sigma \rightarrow 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ is a function mapping each agent $\mu \in \Sigma$ to a set of argument pairs $\mathsf{inv}_{\rightharpoondown}(\mu)$ such that if $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoondown}(\mu)$, then $\mathbf{a} \rightharpoondown \mathbf{b}$ according to $(\mathcal{A}, \rightharpoondown)$, but $\mathbf{a} \rightharpoondown \mathbf{b}$ is an invalid attack with respect to $\mathcal{C}_{\mu}$.

This function serves much the same role as the previous, but focuses on cases where agents have presumed exclusivity between terms not recognised by peers.

A Distributed Portrayal Mechanism

A **portrayal instance** $\mathcal{P}[\sigma]$ used by an agent σ in context $\mathcal{C}$ to describe a portrayal $\mathcal{P}$ of an interaction $\mathcal{I}$ is described by a tuple $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoonup), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoonup}, \mathsf{obs}, \mathsf{acc})$ where:

- The **observation** function $\mathsf{obs} : \Sigma \to 2^{\Gamma}$ is a function mapping each agent $\mu \in \Sigma$ to a set of sentences $\mathsf{obs}(\mu) \subseteq \Gamma$ such that if $\varphi \in \mathsf{obs}(\mu)$, then $\Theta \vdash \varphi$ according to $\mathcal{C}_{\mu}$, where Θ is the theory core of $\mathcal{C}_{\mu}$ and if $\varphi \in \Gamma$, then there must exist an argument $\langle \Phi, \alpha \rangle \in \mathcal{A}$ such that $\Phi \vdash \neg\varphi$ according to $\mathcal{C}_{\mu}$.

This function serves to identify the theory cores of peers, allowing an agent to infer a combined core (if desired).

A Distributed Portrayal Mechanism

A **portrayal instance** $\mathcal{P}[\sigma]$ used by an agent σ in context $\mathcal{C}$ to describe a portrayal $\mathcal{P}$ of an interaction $\mathcal{I}$ is described by a tuple $(\Sigma, \Upsilon, (\mathcal{A}, \rightharpoondown), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$ where:

- The **acceptance** function $\mathsf{acc} : \Sigma \rightarrow 2^{\mathcal{A}}$ 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 $\mathcal{C}_\mu$.

This function serves to identify the preferred interpretations of the portrayal argument system held by peers.

Procedure: argue($\mathcal{P}[\sigma]$)

Given a portrayal instance $\mathcal{P}[\sigma]$, an agent σ in context $\mathcal{C}$ should insert new arguments into $\mathcal{P}$ if and only if:

- There exists at least one 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 $\mathbf{b} \in \mathcal{U}$, the set of unrejected arguments in $\mathcal{C}$.

If this condition is met:

1. Whilst there exists an argument $\mathbf{b} \in \mathcal{U}$ such that:
   - $\mathbf{b} \rightharpoondown \mathbf{c}$ for some elaboration $\mathbf{c} \in \mathcal{C}$ upon an argument $\mathbf{c}' \in \mathcal{P}$ such that $\mathbf{c}' \sqsubseteq \mathbf{c}$.
   - There does not exist an alternative elaboration $\mathbf{d} \in \mathcal{E}$, where $\mathcal{E}$ is the accepted extension of $\mathcal{C}$, upon $\mathbf{c}'$ such that $\mathbf{c}' \sqsubseteq \mathbf{d} \not\sqsubseteq \mathbf{c}$ and $\mathbf{c} \not\sqsubseteq \mathbf{d}$.
   - There does not exist an attack $\mathbf{e} \rightharpoondown \mathbf{c}'$ according to $\mathcal{P}$ already, where $\mathbf{e} \sqsubseteq \mathbf{b}$ and $\mathbf{b} \sqsubseteq \mathbf{e}$.

   Invoke attack($\mathcal{P}[\sigma], \mathbf{e} \rightharpoondown \mathbf{c}'$), where $\mathbf{e} \in \Delta$ and $\mathbf{e} \sqsubseteq \mathbf{b}$.

Procedure: argue($\mathcal{P}[\sigma]$)

Given a portrayal instance $\mathcal{P}[\sigma]$, an agent σ in context $\mathcal{C}$ should insert new arguments into $\mathcal{P}$ if and only if:

- There exists at least one 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 $\mathbf{b} \in \mathcal{U}$, the set of unrejected arguments in $\mathcal{C}$.

If this condition is met:

2. Whilst there exists an argument $\mathbf{b} \in \mathcal{U}$ such that:
   - There does not exist an argument $\mathbf{b}' \in \mathcal{P}$ such that $\mathbf{b}' \sqsubseteq \mathbf{b}$.
   - There does exist an argument $\mathbf{c} \in \Delta$, where Δ is the argument space of $\mathcal{P}$, such that $\mathbf{c} \sqsubseteq \mathbf{b}$.

   Invoke posit($\mathcal{P}[\sigma], \mathbf{c}$).

Operation: posit($\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle$)

An agent σ in context $\mathcal{C}$ with 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 posit($\langle \Phi, \alpha \rangle$) is then dispatched to all agents in Σ of $\mathcal{P}$ (including agent σ).

Function: $\mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle)$

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 $\mathbf{b} \sqsubset \langle \Phi, \alpha \rangle$.
- $\mathcal{P}[\sigma] = (\Sigma, \Upsilon, (\mathcal{A}, \rightharpoondown), \mathsf{inv}_{\mathcal{A}}, \mathsf{inv}_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$.

If these conditions are met, then ...

Function: $\mathcal{P}[\sigma]' = \mathsf{insert}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle)$

$\mathcal{P}[\sigma]' = (\Sigma, \Upsilon, (\mathcal{A}', \rightharpoondown), \mathsf{inv}'_{\mathcal{A}}, \mathsf{inv}'_{\rightharpoondown}, \mathsf{obs}, \mathsf{acc})$, where:

- $\mathcal{A}' = (\mathcal{A}/S) \cup \{\langle \Phi, \alpha \rangle\}$.
- 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 $\mathbf{a} \in \mathsf{inv}'_{\mathcal{A}}(\mu)$.
- If $\mathbf{a} \rightharpoondown \mathbf{b}$, where $\mathbf{a} \in S$ and $\mathbf{b} \in \mathcal{A}$, then $\langle \Phi, \alpha \rangle \rightharpoondown \mathbf{b}$; if $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoondown}(\mu)$ for any peer $\mu \in \Sigma$, then $(\langle \Phi, \alpha \rangle, \mathbf{b} \in \mathsf{inv}'_{\rightharpoondown}(\mu)$.
- If $\mathbf{b} \rightharpoondown \mathbf{a}$, where $\mathbf{a} \in S$ and $\mathbf{b} \in \mathcal{A}$, then $\mathbf{b} \rightharpoondown \langle \Phi, \alpha \rangle$; if $(\mathbf{b}, \mathbf{a} \in \mathsf{inv}_{\rightharpoondown}(\mu)$ for any peer $\mu \in \Sigma$, then $(\mathbf{b}, \langle \Phi, \alpha \rangle) \in \mathsf{inv}'_{\rightharpoondown}(\mu)$.
- Otherwise, if $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}_{\rightharpoondown}(\mu)$, then $(\mathbf{a}, \mathbf{b}) \in \mathsf{inv}'_{\rightharpoondown}(\mu)$ for all peers $\mu \in \Sigma$.

Function: $\mathcal{P}[\sigma]' = \text{insert}(\mathcal{P}[\sigma], \langle \Phi, \alpha \rangle)$

If an elaboration $\langle \Psi, \alpha \rangle \in \mathcal{C}$ of $\langle \Phi, \alpha \rangle$ is dismissed 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 $\langle \Phi, \alpha \rangle \sqsubseteq \langle \Psi'', \alpha \rangle \sqsubset \langle \Psi', \alpha \rangle$ such that $\Psi'' \vdash \neg\varphi$.
- If $\langle \Psi', \alpha \rangle = \langle \Phi, \alpha \rangle$, then invoke observe($\mathcal{P}[\sigma]', \{\varphi\}$).
- Otherwise, invoke elaborate($\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle, \langle \Psi', \alpha \rangle$) and observe($\mathcal{P}[\sigma]', \{\varphi\}$).

If $\langle \Phi, \alpha \rangle$ is invalid with respect to $\mathcal{C}$, then invoke dismiss($\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle$); otherwise invoke reconcile($\mathcal{P}[\sigma]', \langle \Phi, \alpha \rangle$).

Procedure: reconcile($\mathcal{P}[\sigma]$)

Given a portrayal instance $\mathcal{P}[\sigma]$, an agent σ in context $\mathcal{C}$ should reconcile $\mathcal{C}$ with $\mathcal{P}$ if and only if:

- There exists at least one argument $\mathbf{a} \in \mathcal{P}$ such that $\mathbf{a}$ is valid according to $\mathcal{C}$, but there exists no argument $\mathbf{b} \in \mathcal{C}$ such that $\mathbf{a} \sqsubseteq \mathbf{b}$.
- N is the set of all such arguments $\mathbf{a}$.

If these conditions are met, then $\mathcal{C}$ should be updated:

- S is the set of all arguments $\mathbf{a}' \in \mathcal{C}$ such that $\mathbf{a}' \sqsubset \mathbf{a}$ for any argument $\mathbf{a} \in N$.
- If there exist any arguments $\mathbf{a} \in N$ not in Δ of $\mathcal{C}$, then extend Δ to include $\mathbf{a}$.
- Replace $\mathcal{A}$ of $\mathcal{C}$ with the potential expansion of $(\mathcal{A}/S) \cup N$ into Δ.
- If acc(σ) is *not* a potential restriction of $\mathcal{E}$ into Δ of $\mathcal{P}$, then invoke accept($\mathcal{P}[\sigma], \mathcal{E}'$), where $\mathcal{E}'$ *is* a potential restriction of $\mathcal{E}$ into Δ of $\mathcal{P}$.
- Invoke argue($\mathcal{P}[\sigma]$).

A Distributed Portrayal Mechanism

The specification for distributed interaction portrayal ensures that all agents will synchronise their beliefs within an argument space defined by the associated interaction state.

- Provided that the logical frameworks used in their theory contexts can themselves be reconciled.
- Problems of ontology can lead to degradation.
- However invalid arguments can be noted and repaired.

The decoupling of action and response ensures operability in asynchronous environments.

Conclusions

In conclusion:

- Interaction portrayals are used to drive prioritised belief revision on demand.
- By reconciling their theory contexts with the portrayal, agents synchronise their beliefs within the portrayal argument space.
- A distributed process model has been specified which has been proven to ensure reconciliation in response to the interaction state.

Conclusions

Future work:

- Expand study of potential argumentation for full mapping between argumentation frameworks.
- Consider further at least two approaches for merging independent theory contexts into a restricted argument space:
  - Finding the minimal hypothesis space which is expressive enough to sufficiently defend an agent's beliefs from defeatable attacks by peers (the portrayal approach).
  - Finding well-formed abstractions of complex problem domains by the incremental balancing of minimal arguments (developing simpler problem ontologies).