Accountability and Multiagent Organizations: From Concepts to Sotware Engineering

Accountability and Multiagent Organizations

from Concepts to Software Engineering

Matteo Baldoni, Cristina Baroglio, Roberto Micalizio

21st July 2021

Università degli Studi di Torino,
Dipartimento di Informatica

di.unito.it

Accountability?

A constellation of views

- *Social Sciences*: Ethnomethodology, Garfinkel, Rawls & David;
- *Political Sciences*: Anderson, Government of Canada, Grant & Keohane, Melvin Dubnick, Bovens;
- *Tort Law*: Goldberg and Zipursky;
- *Social Psychology*: Tetlock;
- *Philosophy*: Robert Nozick, Stephen Darwall;
- ... *many* ...

Accountability $\rightarrow$ Blame

Accountability and Blame [Dubnick, 2013]

1. **Post factum**: who is to blame for an act or an error that has occurred;

Accountability and Blame [Dubnick, 2013]

1. **Post factum**: who is to blame for an act or an error that has occurred;
2. **Pre factum**: who is blameworthy for errors not yet occurred.

Accountability and Blame [Dubnick, 2013]

1. **Post factum**: who is to blame for an act or an error that has occurred;
2. **Pre factum**: who is blameworthy for errors not yet occurred.

Types of blame cultures:

1. Legalistic
2. Stigma
3. Giri
4. Prejudicial

Moral pushes and pulls

Robert Nozick (philosopher) distinguishes 'moral pulls' from 'moral pushes':

- **moral push**: emphasizes the person who is the subject of a moral life, their character and motivation
- **moral pull**: emphasizes the entities in the world outside of the moral agent as a source of value that generates obligations which exert a pull on the agent

Accountability and Setting [Dubnick, 2013]

Accountability as deriving from the combination of moral pushes and pulls:

Setting	Moral Pulls	Moral Pushes
Legal	Liability	Obligation
Organizational	Answerability	Obedience
Professional	Responsibility	Fidelity
Political	Responsiveness	Amenability

Accountability and Setting [Dubnick, 2013]

Accountability as deriving from the combination of moral pushes and pulls:

Setting	Moral Pulls	Moral Pushes
Legal	Liability	Obligation
Organizational	Answerability	Obedience
Professional	Responsibility	Fidelity
Political	Responsiveness	Amenability

How do these relate to blame?

Accountability and Blame

- **liable**: legally blameworthy (if not satisfying obligation)
- **answerable**: blameworthy (if not obedient)
- **responsible**: be in control so that you will not be blamed by those who trust you
- **responsive**: that adapts (amenable: capable of submission)

Accountability and Setting [Dubnick, 2013]

Accountability as deriving from the combination of moral pushes and pulls:

Setting	Moral Pulls	Moral Pushes
Legal	Liability	Obligation
Organizational	Answerability	Obedience
Professional	Responsibility	Fidelity
Political	Responsiveness	Amenability

Accountability and Setting [Dubnick, 2013]

Accountability as deriving from the combination of moral pushes and pulls:

Setting	Moral Pulls	Moral Pushes
Legal	Liability	Obligation
Organizational	Answerability	Obedience
Professional	Responsibility	Fidelity
Political	Responsiveness	Amenability

ALWAYS INTER-PERSONAL

Accountability and Setting [Dubnick, 2013]

Accountability as deriving from the combination of moral pushes and pulls:

Setting	Moral Pulls	Moral Pushes
Legal	Liability	Obligation
Organizational	Answerability	Obedience
Professional	Responsibility	Fidelity
Political	Responsiveness	Amenability

ALWAYS INTER-PERSONAL

In moral philosophy terms: **second-personal** rather than **first-** (me thinking of myself) or **third-personal** (coming from the outside) [Darwall, 2006]

Example (public administration)

Co-existing accountability systems in the [Romzek and Dubnick, 1987]

- **Bureaucratic**: superior/subordinate relationships, orders unquestioned, close supervision (or standard operating procedures)

Example (public administration)

Co-existing accountability systems in the [Romzek and Dubnick, 1987]

- **Bureaucratic**: superior/subordinate relationships, orders unquestioned, close supervision (or standard operating procedures)
- **Legal**: the lawmaker is an outsider to the organization, the organization executes (fiduciary principal-agent relationship)

Example (public administration)

Co-existing accountability systems in the [Romzek and Dubnick, 1987]

- **Bureaucratic**: superior/subordinate relationships, orders unquestioned, close supervision (or standard operating procedures)
- **Legal**: the lawmaker is an outsider to the organization, the organization executes (fiduciary principal-agent relationship)
- **Professional**: control of the professional activity put in the hands of a skilled employee (manager as layperson, employee as professional, deference to expertise)

Example (public administration)

Co-existing accountability systems in the [Romzek and Dubnick, 1987]

- **Bureaucratic**: superior/subordinate relationships, orders unquestioned, close supervision (or standard operating procedures)
- **Legal**: the lawmaker is an outsider to the organization, the organization executes (fiduciary principal-agent relationship)
- **Professional**: control of the professional activity put in the hands of a skilled employee (manager as layperson, employee as professional, deference to expertise)
- **Political**: constituent/representative relationship, responsiveness to constituents.

Example (public administration)

Co-existing accountability systems in the [Romzek and Dubnick, 1987]

- **Bureaucratic**: superior/subordinate relationships, orders unquestioned, close supervision (or standard operating procedures)
- **Legal**: the lawmaker is an outsider to the organization, the organization executes (fiduciary principal-agent relationship)
- **Professional**: control of the professional activity put in the hands of a skilled employee (manager as layperson, employee as professional, deference to expertise)
- **Political**: constituent/representative relationship, responsiveness to constituents.

Sometimes seen as systems for managing expectations.

Accountability: a chamelion word [Dubnick and Justice, 2004]

Many understandings ...

Accountability ...

Accountability: a chamelion word [Dubnick and Justice, 2004]

Many understandings ...

- that show the same relatedness shown by individuals from a same family

Accountability ...

Accountability: a chamelion word [Dubnick and Justice, 2004]

Many understandings ...

- that show the same relatedness shown by individuals from a same family

Accountability ...

- "emerges as a **primary characteristic of governance** where there is a sense of agreement and certainty about the legitimacy of expectations between the community members."

Accountability: a chamelion word [Dubnick and Justice, 2004]

Many understandings ...

- that show the same relatedness shown by individuals from a same family

Accountability ...

- "emerges as a primary characteristic of governance where there is a sense of **agreement and certainty** about the **legitimacy of expectations** between the community members."

[Grant and Keohane, 2005]

- "Accountability, as we use the term, implies that *some actors have the right to hold other actors to a set of standards*, to judge whether they have fulfilled their responsibilities in light of these standards, and to impose sanctions if they determine that these responsibilities have not been met."

[Grant and Keohane, 2005]

- "Accountability, as we use the term, implies that *some actors have the right to hold other actors to a set of standards*, to judge whether they have fulfilled their responsibilities in light of these standards, and to impose sanctions if they determine that these responsibilities have not been met."
- "Accountability presupposes a relationship between power-wielders and those holding them accountable where there is a general recognition of the legitimacy of (1) the operative standards for accountability and (2) the authority of the parties to the relationship (one to exercise particular powers and the other to hold them to account). "

Grant & Keohane [Grant and Keohane, 2005]

- "Accountability, as we use the term, implies that *some actors have the right to hold other actors to a set of standards*, **to judge whether they have fulfilled their responsibilities** in light of these standards, and to impose sanctions if they determine that these responsibilities have not been met."
- "Accountability presupposes a relationship between power-wielders and those holding them accountable where there is a general recognition of the legitimacy of (1) the operative standards for accountability and (2) the authority of the parties to the relationship (one to exercise particular powers and the other to hold them to account). "

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

1. **Virtue**: Smith had always been an exceedingly responsible person,

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

1. **Virtue**: Smith had always been an exceedingly responsible person,
2. **Role**: and as captain of the ship he was responsible for the safety of his passengers and crew.

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

1. **Virtue**: Smith had always been an exceedingly responsible person,
2. **Role**: and as captain of the ship he was responsible for the safety of his passengers and crew.
3. **Outcome**: But on his last voyage he drank himself into a stupor, and he was responsible for the loss of his ship and many lives.

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

1. **Virtue**: Smith had always been an exceedingly responsible person,
2. **Role**: and as captain of the ship he was responsible for the safety of his passengers and crew.
3. **Outcome**: But on his last voyage he drank himself into a stupor, and he was responsible for the loss of his ship and many lives.
4. **Causal**: Smith's defense attorney argued that the alcohol and his transient depression were responsible for his misconduct,

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

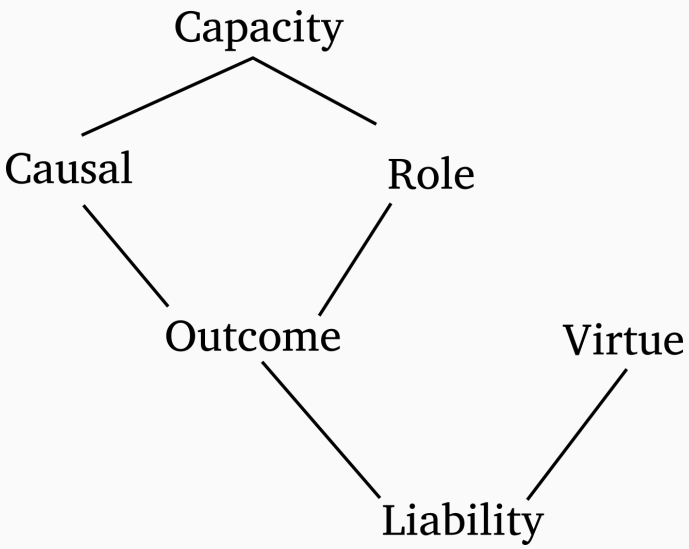
1. **Virtue**: Smith had always been an exceedingly responsible person,
2. **Role**: and as captain of the ship he was responsible for the safety of his passengers and crew.
3. **Outcome**: But on his last voyage he drank himself into a stupor, and he was responsible for the loss of his ship and many lives.
4. **Causal**: Smith's defense attorney argued that the alcohol and his transient depression were responsible for his misconduct,
5. **Capacity**: but the prosecution's medical experts confirmed that he was fully responsible when he started drinking since he was not suffering from depression at that time.

Parenthetical: Responsibility

Kinds of responsibility [Vincent, 2011], from *Smith the ship captain*, by philosopher H.L.A. Hart:

1. **Virtue**: Smith had always been an exceedingly responsible person,
2. **Role**: and as captain of the ship he was responsible for the safety of his passengers and crew.
3. **Outcome**: But on his last voyage he drank himself into a stupor, and he was responsible for the loss of his ship and many lives.
4. **Causal**: Smith's defense attorney argued that the alcohol and his transient depression were responsible for his misconduct,
5. **Capacity**: but the prosecution's medical experts confirmed that he was fully responsible when he started drinking since he was not suffering from depression at that time.
6. **Liability**: Smith should take responsibility for his victims' families' losses, but his employer will probably be held responsible for them as Smith is insolvent and uninsured.

Ontology of Responsibilities



Parenthetical: Responsibility

In MAS literature:

- "one being responsible for a task" is understood as "the one who carries out the task" (survey [Feltus, 2014], see also [Yazdanpanah and Dastani, 2016])
- Goal decomposition and distribution (e.g. [Boissier et al., 2013])
- In [Baldoni et al., 2019b] we see a responsibility as an agent being "a recipient" for (and being moved by) some institutional event

Back to Accountability

- “Accountability presupposes a relationship between power-wielders and those holding them accountable where there is a general recognition of the legitimacy of (1) the operative standards for accountability and (2) the authority of the parties to the relationship (one to exercise particular powers and the other to hold them to account). ”

Why at all accounting for something?

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]
- **Sanction for infringement**: 20 million euros

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino
  - ... which is divided into 26 departments

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino
  - ... which is divided into 26 departments
  - In case of infringement, no matter what, the Dean is liable

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016 [European Commission, 2018]
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino
  - ... which is divided into 26 departments
  - In case of infringement, no matter what, the Dean is liable

OBLIGATIONS/SANCTIONS

(Blame culture approach)

Organization Engineering Problems

- **Lack of capability**: an agent who does not have the capability to do something will not do it even if obliged (and sanctioned upon failure);
- **Convenience**: a rational agent that finds a sanction more acceptable than satisfying an obligation to do a task, that does not comply with the agent's goals, will not abide by the obligation (and will not explain the reasons).

Organization Engineering Problems

- **Lack of capability**: an agent who does not have the capability to do something will not do it even if obliged (and sanctioned upon failure);
- **Convenience**: a rational agent that finds a sanction more acceptable than satisfying an obligation to do a task, that does not comply with the agent's goals, will not abide by the obligation (and will not explain the reasons).

Blame is not enogh:

Sanction does not add capability nor it increases the responsabilization of the agents.

Well-known in Sociology

[Durkheim, 1893], [Parsons, 1968], [Garfinkel, 1967], etc.

- obligation insufficient to explain social action,
- an agent acts **voluntarily** if the act is **desirable** for the agent
- Normative sanction often has little consequence on the agent and no consequence at the society level

Well-known in Sociology

[Durkheim, 1893], [Parsons, 1968], [Garfinkel, 1967], etc.

- obligation insufficient to explain social action,
- an agent acts **voluntarily** if the act is **desirable** for the agent
- Normative sanction often has little consequence on the agent and no consequence at the society level

Autonomy demands a different way of conceptualizing software modularity:

- Software modularized in terms of subgoals that are assigned to the agents
- Subgoals seen as **responsibilities**

Well-known in Sociology

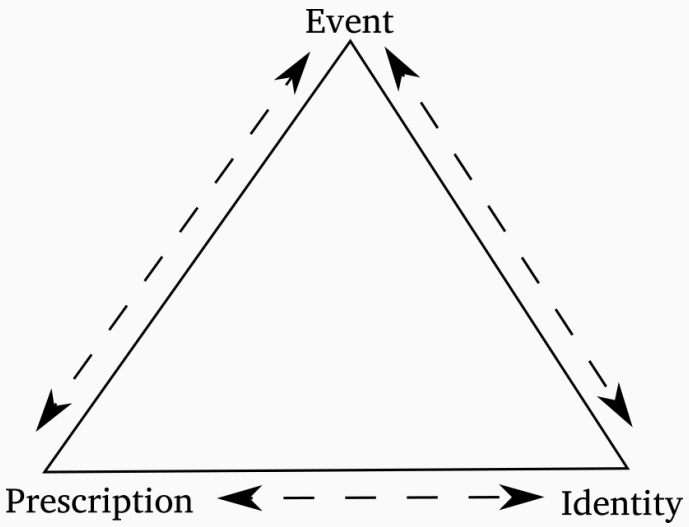
[Durkheim, 1893], [Parsons, 1968], [Garfinkel, 1967], etc.

- obligation insufficient to explain social action,
- an agent acts **voluntarily** if the act is **desirable** for the agent
- Normative sanction often has little consequence on the agent and no consequence at the society level

Autonomy demands a different way of conceptualizing software modularity:

- Software modularized in terms of subgoals that are assigned to the agents
- Subgoals seen as **responsibilities**
- Little problem ...

Triangle Model of responsibility [Schlenker et al., 1994]

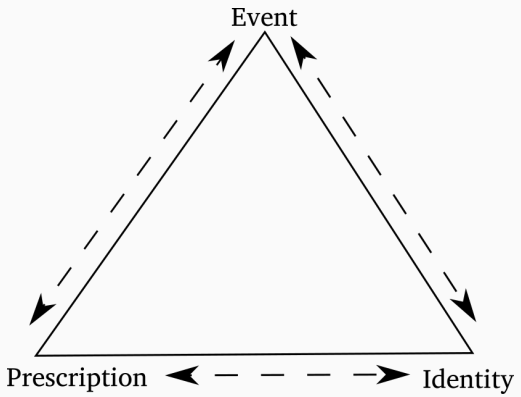


Schlenker et al.

An individual perceives a responsibility when the links are strong: identity-event, event-prescription, prescription-identity.

INSIDE THE AGENT

Triangle Model of responsibility: Example



identity: Luca the doorman,
prescription: should open the door,
event: the bell rings.

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino
  - ... which is divided into 26 departments
  - In case of infringement, no matter what, the Dean is liable

Departments feel responsible

Each Dept. verifies compliance

Answer is: YES!

A student complains to the Dean about some private data being exposed by a Department

A student complains to the Dean about some private data being exposed by a Department

The Dean is blamed (and sanctioned)

And then?

A student complains to the Dean about some private data being exposed by a Department

The Dean is blamed (and sanctioned)

And then?

- What did Departments actually verify? How did they?

A student complains to the Dean about some private data being exposed by a Department

The Dean is blamed (and sanctioned)

And then?

- What did Departments actually verify? How did they?
- Who to talk to inside the involved Department?
- On the basis on which authority asking to someone? General purpose Dean's authority?

A student complains to the Dean about some private data being exposed by a Department

The Dean is blamed (and sanctioned)

And then?

- What did Departments actually verify? How did they?
- Who to talk to inside the involved Department?
- On the basis on which authority asking to someone? General purpose Dean's authority?
- How to gather information for solving the problem and avoiding similar situations in the future?
- A lot of information about the Department's organization is available but the one we need is hidden and must be found.

- Lack of an adequate representation
- Accountability hidden into some kind of collective responsibility – sometimes called “many hands problem”.
- Governance of the system and its functioning as a whole are compromised.

Punishment vs Remedy
Tort Law [Goldberg and Zipursky, 2010]

Legal wrong: VIOLATION OF A DIRECTIVE

Punishment vs Remedy
Tort Law [Goldberg and Zipursky, 2010]

Legal wrong: VIOLATION OF A DIRECTIVE

Criminal Law

- **simple directive**:
  For all x, x shall not A
- empowers the state to hold wrongdoers accountable
- **Accountability**$\rightarrow$ **punishment**

Punishment vs Remedy
Tort Law [Goldberg and Zipursky, 2010]

Legal wrong: VIOLATION OF A DIRECTIVE

Criminal Law

- **simple directive**:
  For all x, x shall not A
- empowers the state to hold wrongdoers accountable
- **Accountability**$\rightarrow$ **punishment**

Tort Law

- **relational directive**:
  For all x and for all y, x shall not do A to y
- **empowers private parties** to initiate proceedings designed to hold tortfeasors accountable
- **Accountability**: the successful victim will have the right to exact a remedy, and courts will apply **principles of remedy**

Responsibility is not enough

Something is missing

Ethnomethodology, a Radical View, H. Garfinkel [Garfinkel, 1967]

Distinctive feature of Garfinkel's approach to social order:

"people organize their actions and interactions as concerted by making them 'accountable' - that is, reciprocally recognizable. Thus, social activities are performed as observable and reportable phenomena. [...]

Ethnomethodology, a Radical View, H. Garfinkel [Garfinkel, 1967]

Distinctive feature of Garfinkel's approach to social order:

"people organize their actions and interactions as concerted by making them 'accountable' - that is, reciprocally recognizable. Thus, social activities are performed as observable and reportable phenomena. [...]

Garfinkel's notion of 'accountability' ... refers to the ways in which actions are organized: that is, put together as publicly observable, reportable occurrences. [...] They are done so that they can be seen to have been done. " [Button and Sharrock, 1998]

Ethnomethodology, a Radical View, H. Garfinkel [Garfinkel, 1967]

Distinctive feature of Garfinkel's approach to social order:

"people organize their actions and interactions as concerted by making them 'accountable' - that is, reciprocally recognizable. Thus, social activities are performed as observable and reportable phenomena. [...]

Garfinkel's notion of 'accountability' ... refers to the ways in which actions are organized: that is, put together as publicly observable, reportable occurrences. [...] They are done so that they can be seen to have been done. " [Button and Sharrock, 1998]

WHY?

A student complains to the Dean ...

- What did Departments actually verify? How did they?

Action is not devised so as to be reportable.

A student complains to the Dean ...

- What did Departments actually verify? How did they?
- Who to talk to inside the involved Department?
- On the basis on which authority asking to someone?

Action is not devised so as to be reportable.
Agents do not share the same conception of legitimacy.

A student complains to the Dean ...

- What did Departments actually verify? How did they?
- Who to talk to inside the involved Department?
- On the basis on which authority asking to someone?
- How to gather information for solving the problem and avoiding similar situations in the future?

Action is not devised so as to be reportable.
Agents do not share the same conception of legitimacy.
Information is hidden, not always accessible.

Let's introduce Accountability

Key Aspects [Baldoni et al., 2019a]

1. *Accountability implies agency.*
   Without the qualities to act "autonomously, interactively and adaptively," i.e. with agency, there is no reason to speak of accountability because we would be talking of a tool, and tools cannot be held accountable [Simon, 2015].

Key Aspects [Baldoni et al., 2019a]

1. *Accountability implies agency.*
   Without the qualities to act "autonomously, interactively and adaptively," i.e. with agency, there is no reason to speak of accountability because we would be talking of a tool, and tools cannot be held accountable [Simon, 2015].
2. *Accountability requires but is not limited to causal significance.*
   The plain, physical causation [Burgemeestre and Hulstijn, 2015, Chopra and Singh, 2014], that does not involve awareness or choice, does not create responsibility nor accountability.

Key Aspects [Baldoni et al., 2019a]

1. *Accountability implies agency.*
   Without the qualities to act "autonomously, interactively and adaptively," i.e. with agency, there is no reason to speak of accountability because we would be talking of a tool, and tools cannot be held accountable [Simon, 2015].
2. *Accountability requires but is not limited to causal significance.*
   The plain, physical causation [Burgemeestre and Hulstijn, 2015, Chopra and Singh, 2014], that does not involve awareness or choice, does not create responsibility nor accountability.
3. *Accountability does not hinder autonomy.*
   It makes sense because of autonomy in deliberation [Anderson, 1981, Schlenker et al., 1994, Suchman, 1997, Chopra and Singh, 2014].

4. *Accountability requires control.*
   Control is the capability, possibly exercised indirectly via other agents, of bringing about events [Marengo et al., 2011] (omissions, i.e. not acting, can be seen as non-achievements).
5. *Accountability requires observability.*
   In order to make correct judgments, a forum must be able to observe the necessary relevant information.

7. *Accountability requires a mutually held expectation.*
   It is a directed social relationship that serves the purposes of sense-making and coordination in a group of interacting parties, all of whom share an agreement on how things should be done [Garfinkel, 1967, Suchman, 1997, Anderson, 1981].

   Both parties must be aware of such a relationship.

8. *Accountability is rights-driven.*
   One is held accountable by another who, in a certain context, has the claim-right to ask for the account [Darwall, 2013, Grant and Keohane, 2005].

Example: Can the Dean sleep quiet dreams?

- **GDPR**: EU Law, General Data Protection Regulation, 2016
- **Sanction for infringement**: 20 million euros
- GDPR affects the University of Torino
  - ... which is divided into 26 departments
  - In case of infringement, no matter what, the Dean is liable

Responsibilization through Accountability

- **Explicitly represent**: who is accountable of what and towards whom, and conditions of the claim-right;
- **Legitimacy**: Agents accept accountabilities.

A student complains ...

A student complains ...

- What did Departments actually verify? How did they?
  The Dean requests a proof

A student complains ...

- What did Departments actually verify? How did they?
  The Dean requests a proof
- Who to talk to inside the involved Department?
  A person designated to be the account-giver

A student complains ...

- What did Departments actually verify? How did they?
  The Dean requests a proof
- Who to talk to inside the involved Department?
  A person designated to be the account-giver
- On the basis on which authority asking to someone?
  The claim-right of the Dean, that the account-giver accepted and of which is aware

A student complains ...

- What did Departments actually verify? How did they?
  The Dean requests a proof
- Who to talk to inside the involved Department?
  A person designated to be the account-giver
- On the basis on which authority asking to someone?
  The claim-right of the Dean, that the account-giver accepted and of which is aware
- How to gather information for solving the problem and avoiding similar situations in the future?
  By requesting the proof

A student complains ...

- What did Departments actually verify? How did they?
  The Dean requests a proof
- Who to talk to inside the involved Department?
  A person designated to be the account-giver
- On the basis on which authority asking to someone?
  The claim-right of the Dean, that the account-giver accepted and of which is aware
- How to gather information for solving the problem and avoiding similar situations in the future?
  By requesting the proof
- It is always clear and accepted who should return accounts to whom and when: sort of additional explicit "infrastructure"

Accountability Dimensions

Normative dimension

it creates mutual expectations on the behavior of the involved agents; it captures the *legitimacy*, for the account taker, of asking (and the *availability* of the account giver to provide) an account (the standing of the account taker to demand an account).

Structural dimension

it concerns the *capability* to produce an account; for being held to account about a process, an agent must exert control over the same process and must have proper awareness of the situation it accounts for, possibly by relying on other agents.

Organization Engineering?

Accountability acceptance exposes the responsibilities agents perceive (previously hidden):

- enables reasoning
- increases system robustness

On legitimate requests:

- **Lack of capability**: the agent will either not play the role or will explain its lack of skill when asked;
- **Convenience**: agents will explain the conflict between their goal and the assigned task.
- **Behave up to the standard**: agents can be asked proofs also when goals are achieved!
  Certification, when "how things are done" matters.

From Blame to Self-regulation

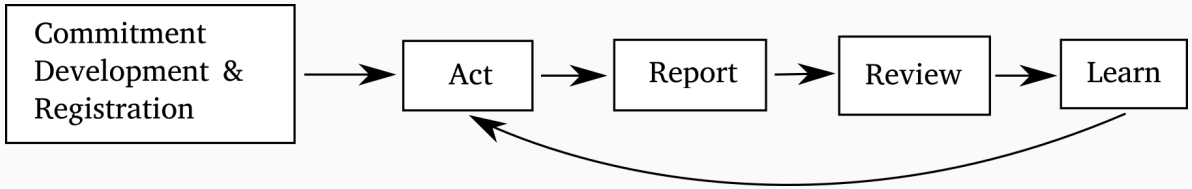


Figure 1: A general scheme for accountability frameworks inspired by [Auditor General of Canada , 2002], appeared in [Baldoni et al., 2018d].

The Power of Accounts

Account is more constructive than blame

Accountability for robustness

Robustness: an important property of software systems

Sys. and Soft. Eng. Vocabulary ISO/IEC/IEEE 24765

Robustness as the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions.

Robustness refers to a system property

A property of a system is robust if it is invariant with respect to a set of perturbations [Alderson and Doyle, 2010].

- **reliability** as robustness to component failure
- **efficency** as robustness to lack of resources
- **scalability** as robustness to change to the size and complexity of the system as a whole
- **modularity** as robustness to structured component rearrangements
- **evolvability** as robustness of lineages to changes on long time scales

Robustness: the role of feedback

The availability of feedback is seen as crucial in gaining robustness [Alderson and Doyle, 2010].

Feedback

A piece of information, some facts that are obtained retroactively, that objectively concern an execution of interest, and that are passed from one component to another.

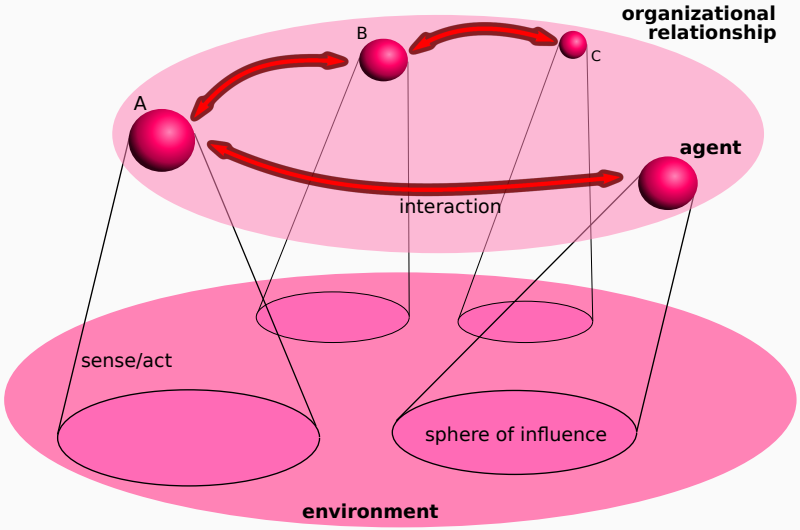
Significance and quality of feedback

are crucial in making a system robust: [Alderson and Doyle, 2010].

- only information that is functional to the desired kind of robustness
- only information that comes from reliable source

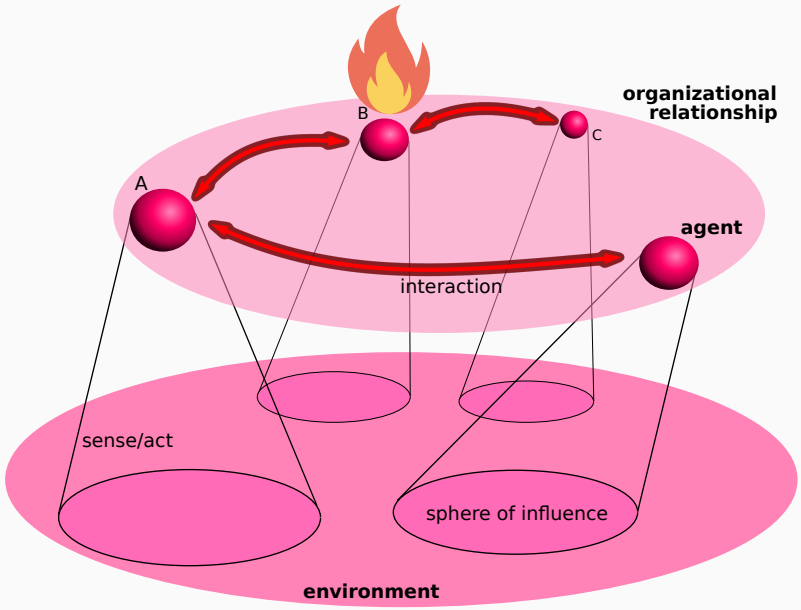
Multiagent Organizations

- "*An organization provides a structure of constraints that allow a system consisting of many parts to act as a whole, with the aim of achieving goals that otherwise would not be achievable (or not as easily)*" [Elder-Vass, 2011]
- **Norms** (rules, protocols, etc.) to define what is expected of each agent
- **Sanctions** as deterrents **to prevent** norm violation



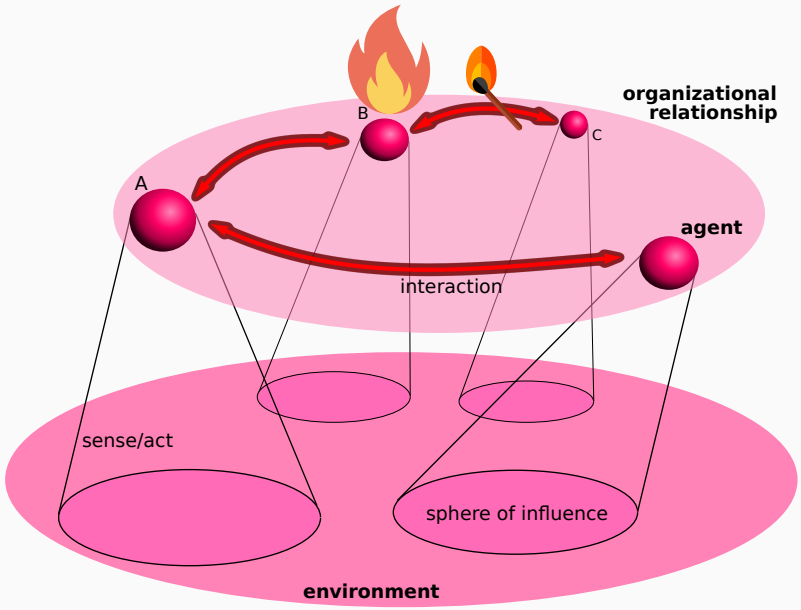
Perturbations

- Unfortunately, when the system faces an abnormal situation (**perturbation**) and some agent *fails* to achieve a goal, **sanctions are of little utility**, if any



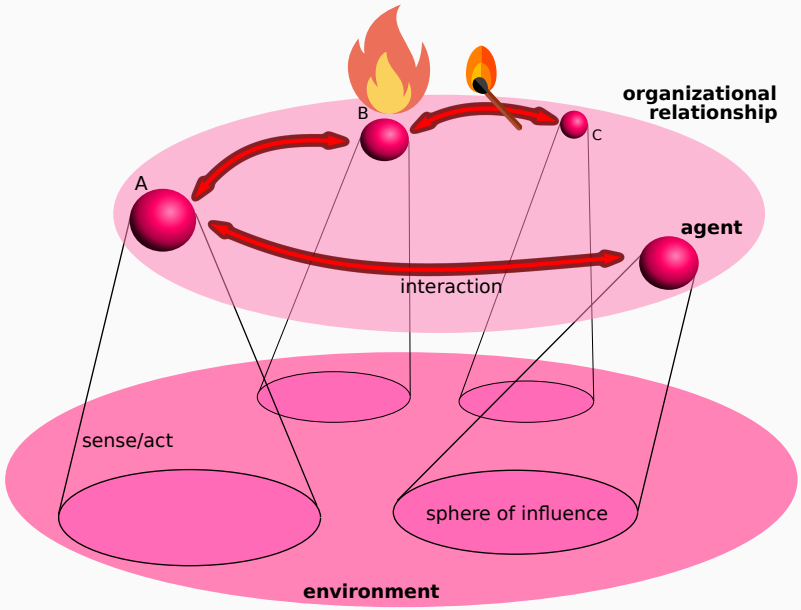
Perturbations

- Unfortunately, when the system faces an abnormal situation (**perturbation**) and some agent *fails* to achieve a goal, **sanctions are of little utility**, if any
- The agent may have tried its best to do what expected, but something which is not under its control might hinder the achievement



Perturbations

- Unfortunately, when the system faces an abnormal situation (**perturbation**) and some agent *fails* to achieve a goal, **sanctions are of little utility**, if any
- The agent may have tried its best to do what expected, but something which is not under its control might hinder the achievement



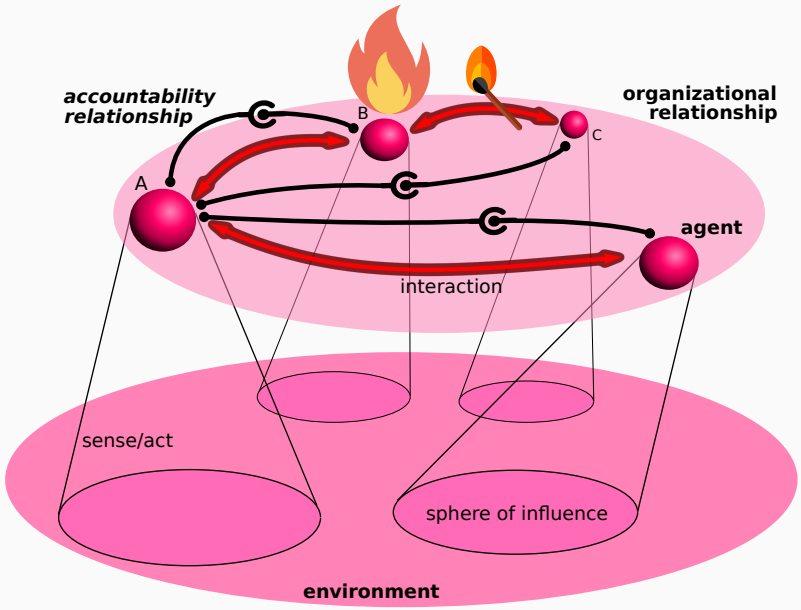
Broader problem

No structured way for collecting and propagating information about encountered situations

Robustness

When a system meets a perturbation it needs **to reconfigure**. To this aim:

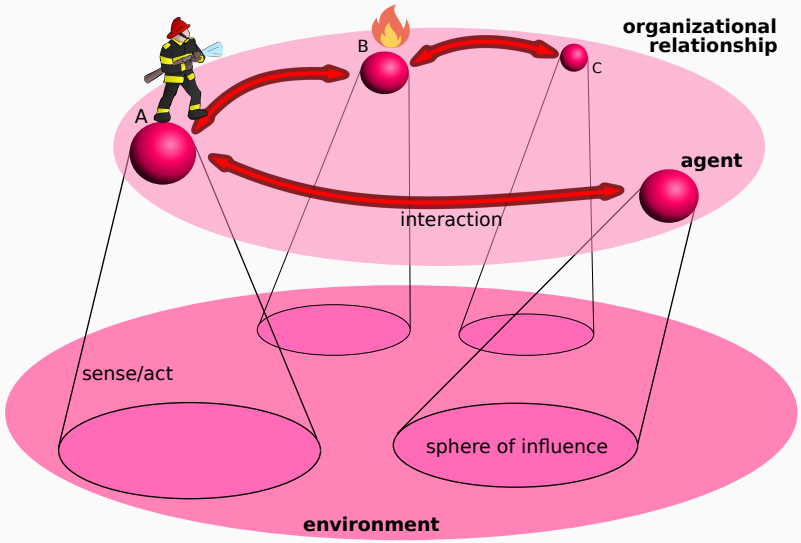
1. the agents need the means for: understanding who is entitled to ask what to whom
2. the information of interest must be asked to an informed source and must be delivered in the right format
3. the information will be delivered to whom is equipped with the right abilities and will be entitled to perform certain tasks, needed to cope with the situation



Robustness

When a system meets a perturbation it needs **to reconfigure**. To this aim:

1. the agents need the means for: understanding who is entitled to ask what to whom
2. the information of interest must be asked to an informed source and must be delivered in the right format
3. the information will be delivered to whom is equipped with the right abilities and will be entitled to perform certain tasks, needed to cope with the situation



Fragility in MAS: the role of feedback

- The **agents' autonomy** is an enabler of the system's adaptability, which is crucial to achieve robustness
  - However, adaptability requires the system to be equipped with the ability to produce proper feedback, propagate it, and process it, so to enable the selection and enactment of behavior that is appropriate to cope with the situation
- The **normative system** enables the exploitation of the agents' autonomy, creating expectations on their activities, which is crucial to achieve system robustness
  - However, agents may fail the expectations (the obligations). Whenever sanctions are not accompanied by feedback and feedback handling mechanisms, they do not provide a means that support robustness

Accountability as a means for robustness in MAS

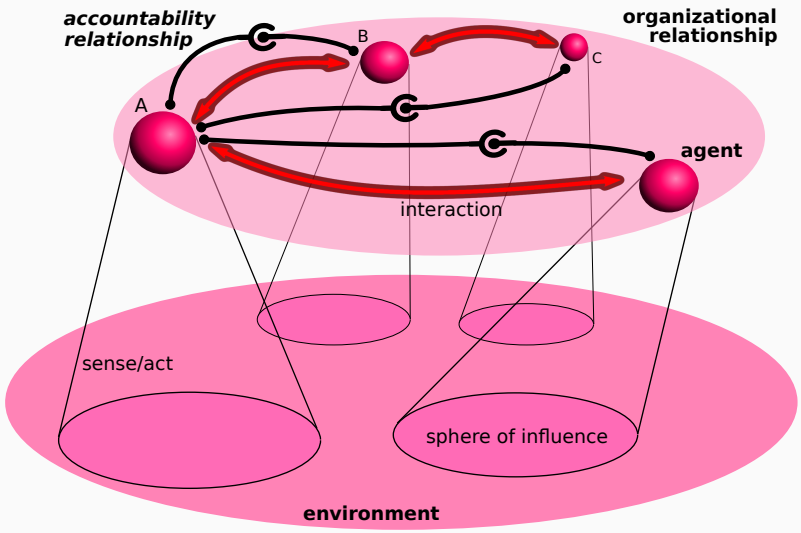
- The current design methodologies for MAS fall short in addressing robustness in a systematic way at design time.

Accountability

We exploit the notion of **accountability** [Garfinkel, 1967, Grant and Keohane, 2005, Dubnick and Justice, 2004, Baldoni et al., 2016, Baldoni et al., 2019a] as a mechanism for building feedback/reporting frameworks, similarly to what is often done in human organizations [Sustainable Energy for All Initiative, , Zahran, 2011].

Accountability as a means for robustness in MAS

- We claim that **account** and **accountability** are the crucial tools for making organizations more **robust**
- In the **human world/organizations** accountability it provides the means to address recurring and systemic issues, and to incorporate lessons learned into future activities



A conceptual model for the accountability

Conceptual Model

A **conceptual model** is a visual representation of conceptual classes or real-situation objects in a domain.

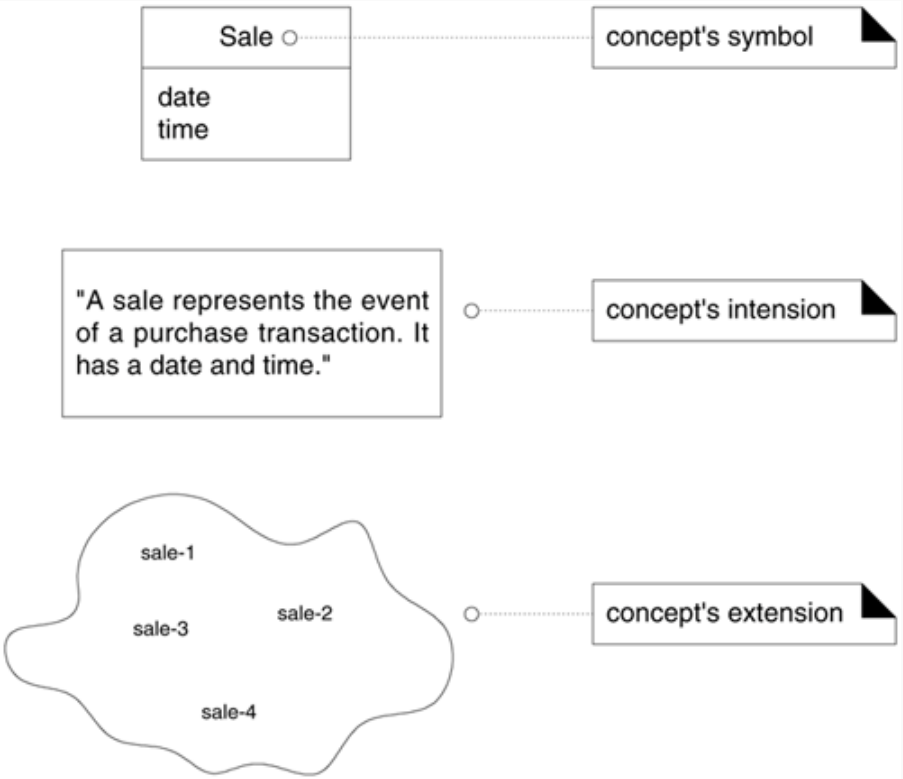
Applying UML notation, a conceptual model is illustrated with a set of class diagrams in which no operations (method signatures) are defined.

It provides a conceptual perspective. It may show:

- **conceptual classes**
- **associations** between conceptual classes
- **attributes** of conceptual classes

Conceptual class

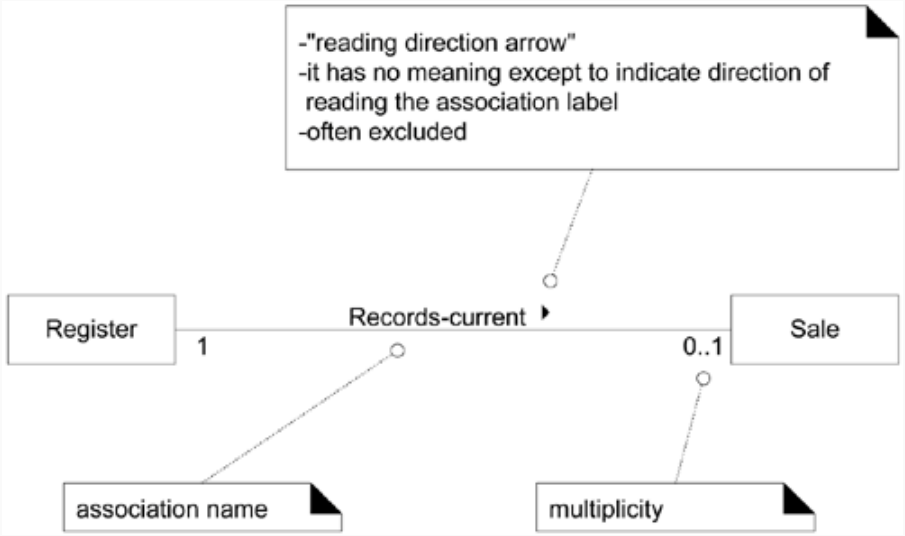
- **Symbol** words or images representing a conceptual class
- **Intension** the definition of a conceptual class
- **Extension** the set of examples to which the conceptual class applies.



©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

Association

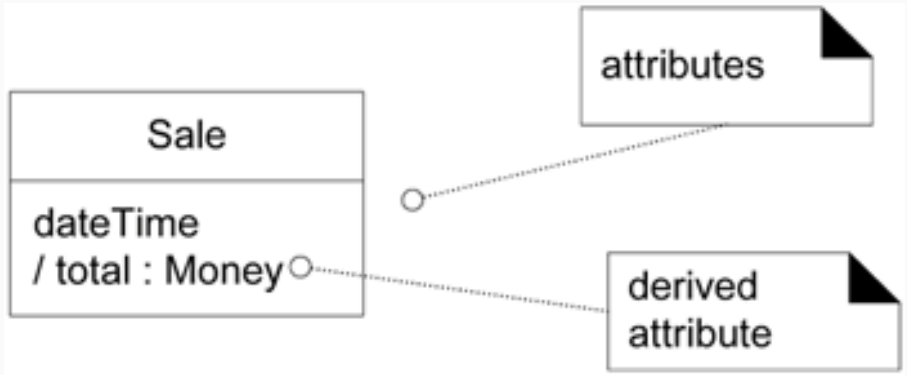
An association is a **relationship** between classes (more precisely, instances of those classes) that indicates some meaningful and interesting connection.



©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

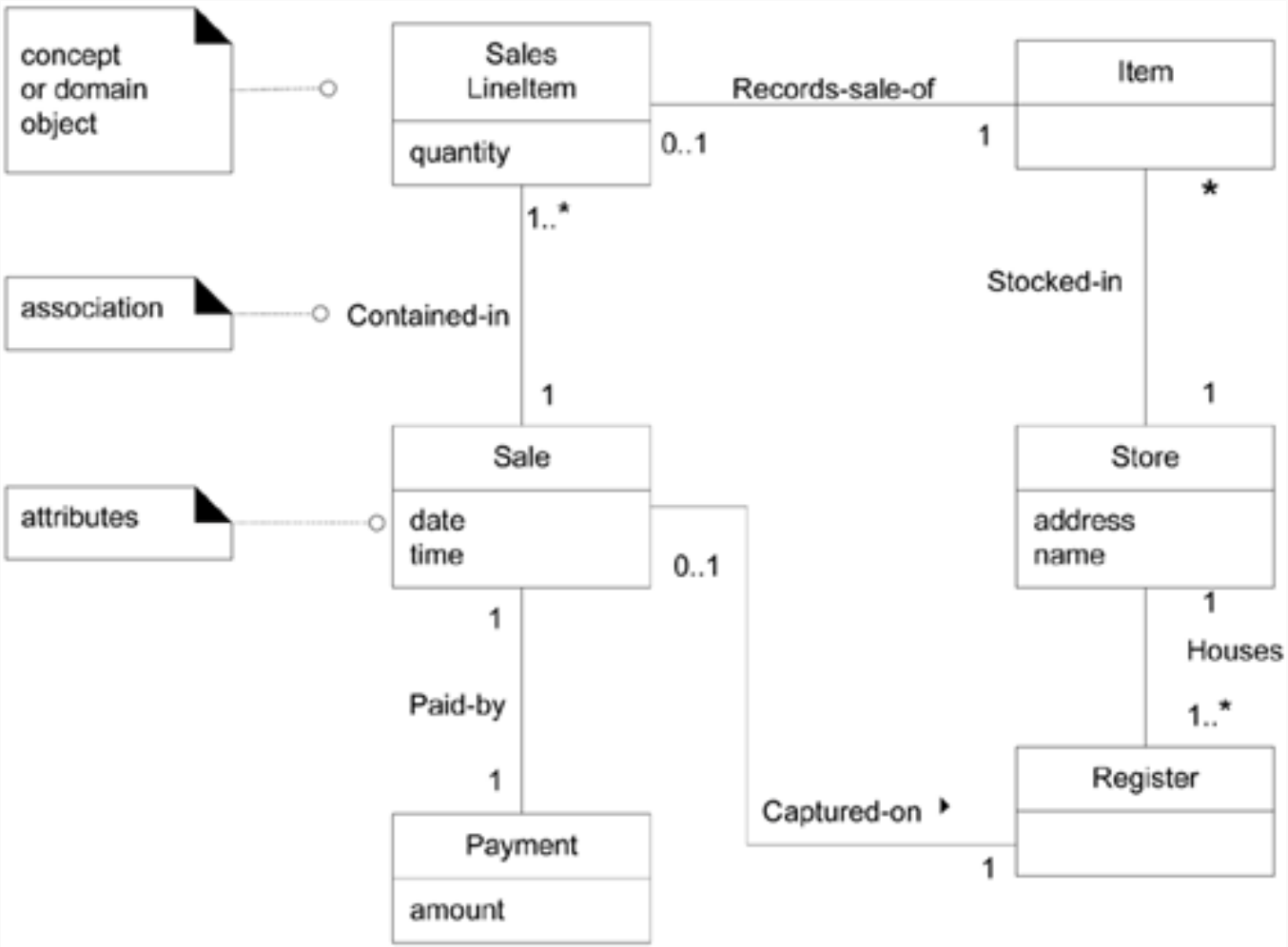
Attribute

An attribute is a logical data value of an object.



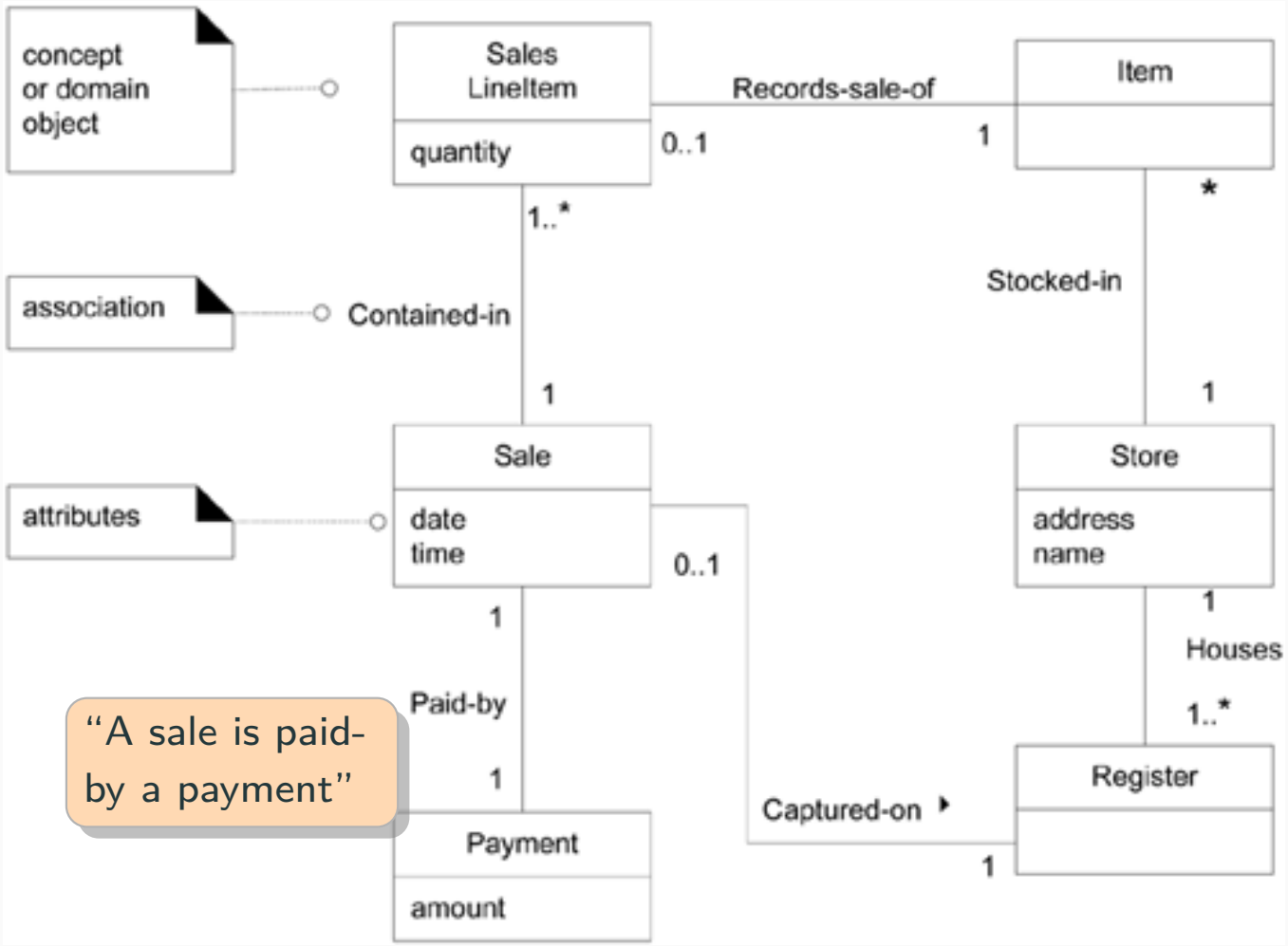
©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

An example of conceptual model



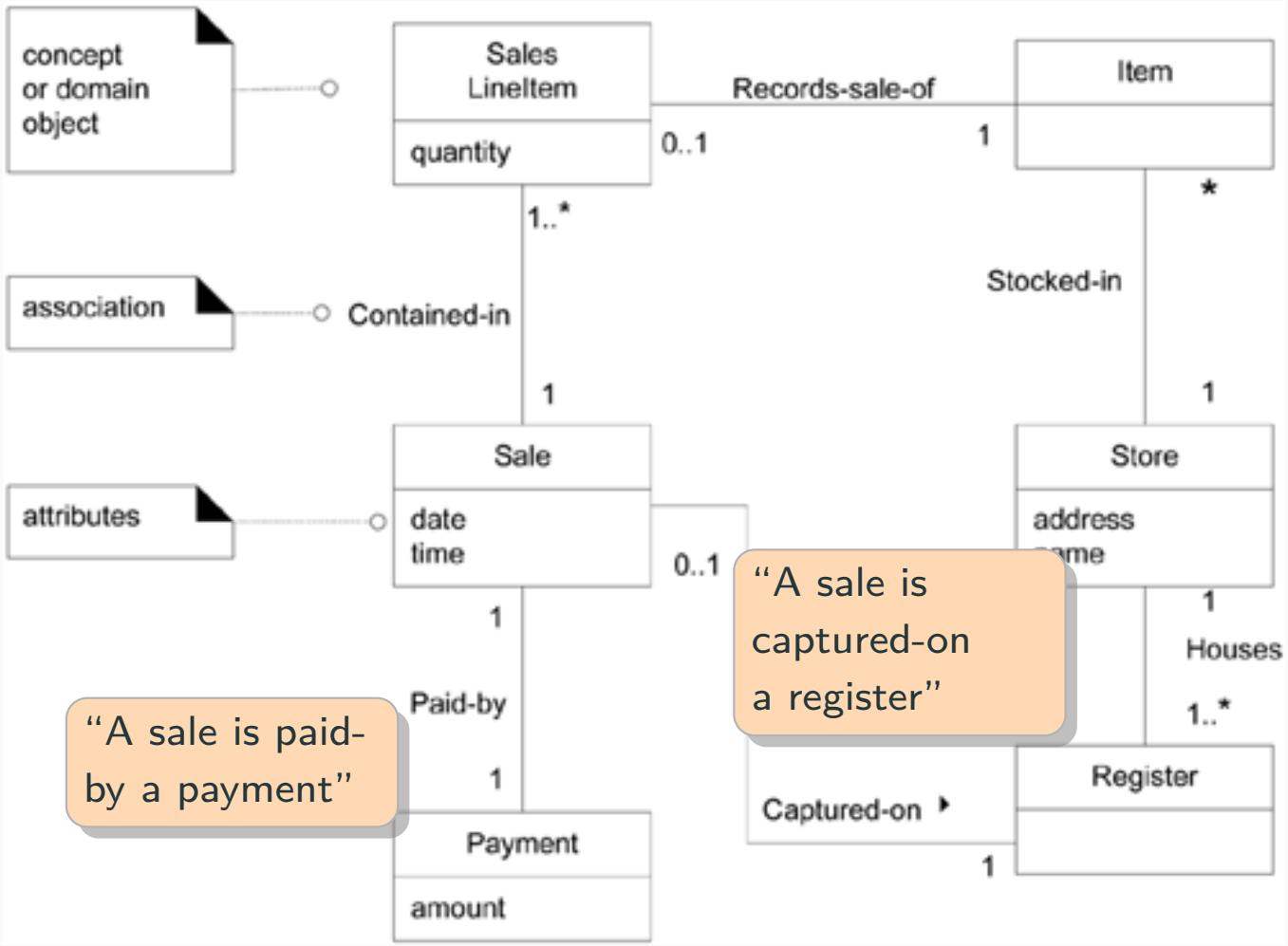
©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

An example of conceptual model



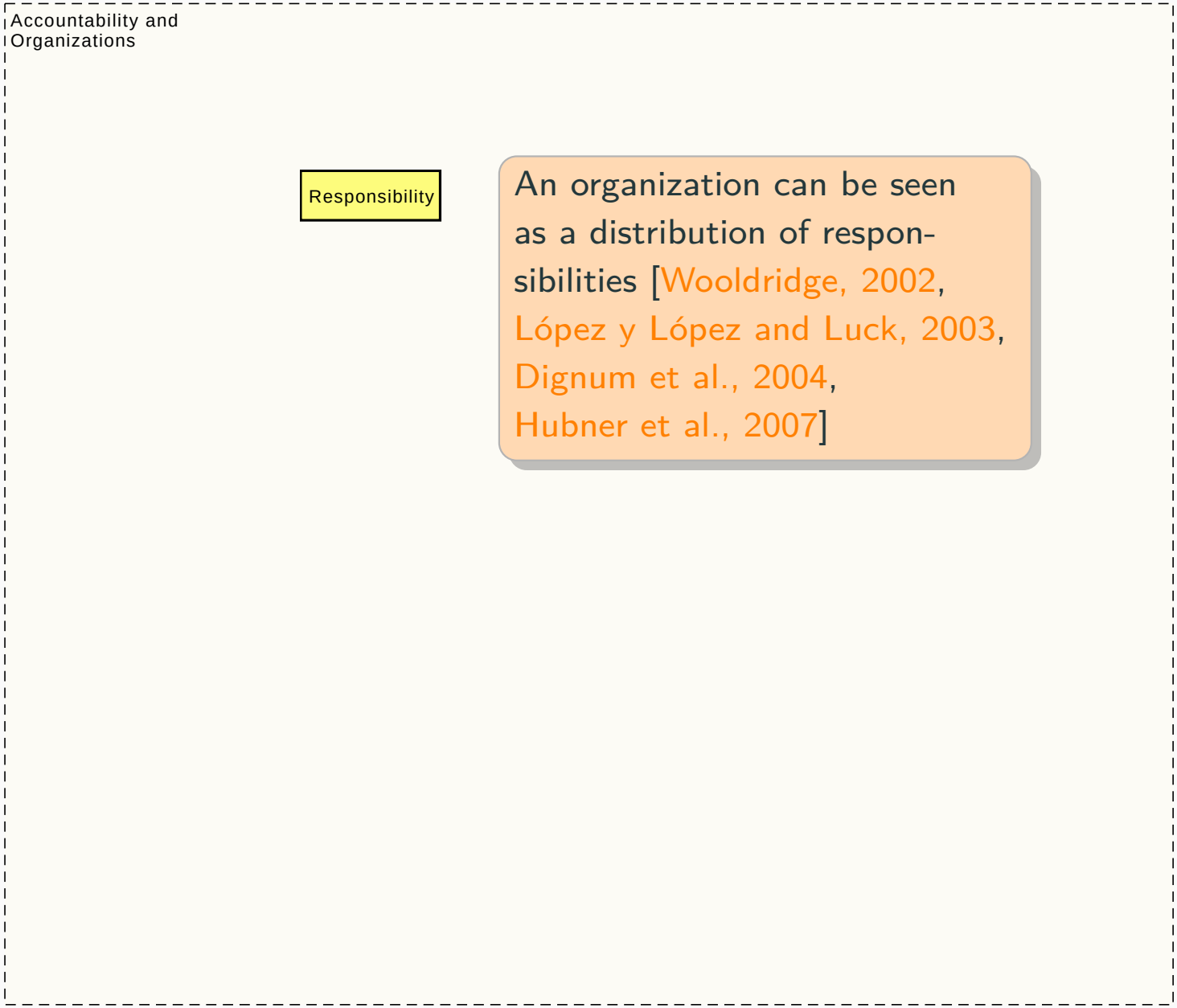
©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

An example of conceptual model

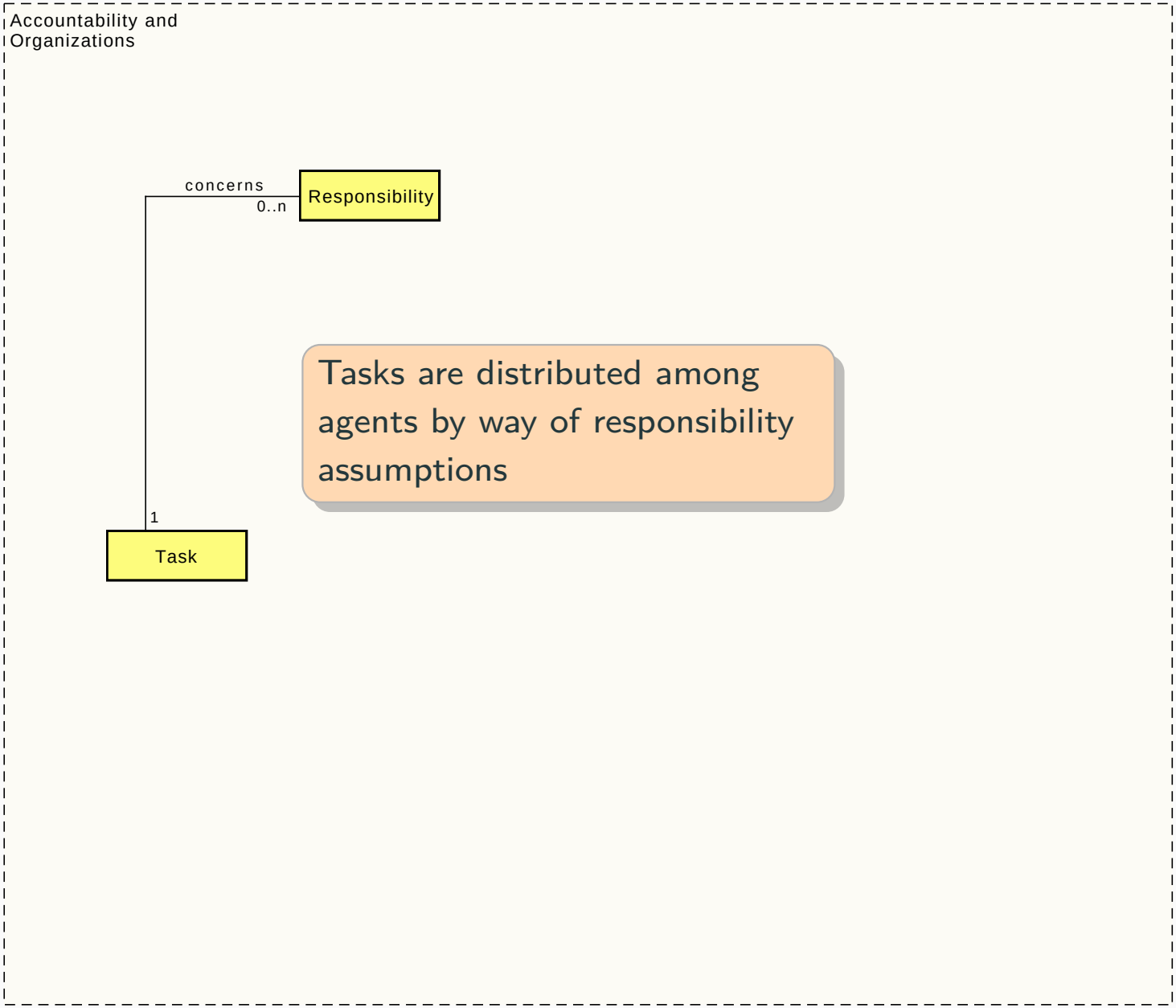


©C. Larman. Applying UML and patterns. Addison Wesley Professional, 2004.

Task, Agent, and Responsibility

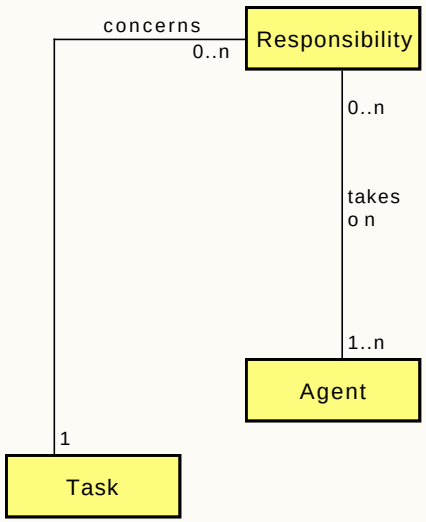


Task, Agent, and Responsibility



Task, Agent, and Responsibility

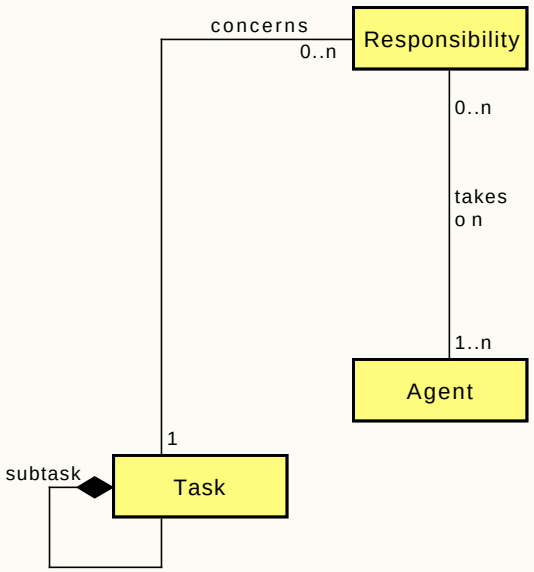
Accountability and Organizations



An agent is part of the organization only when it explicitly takes on the responsibility that concerns some task

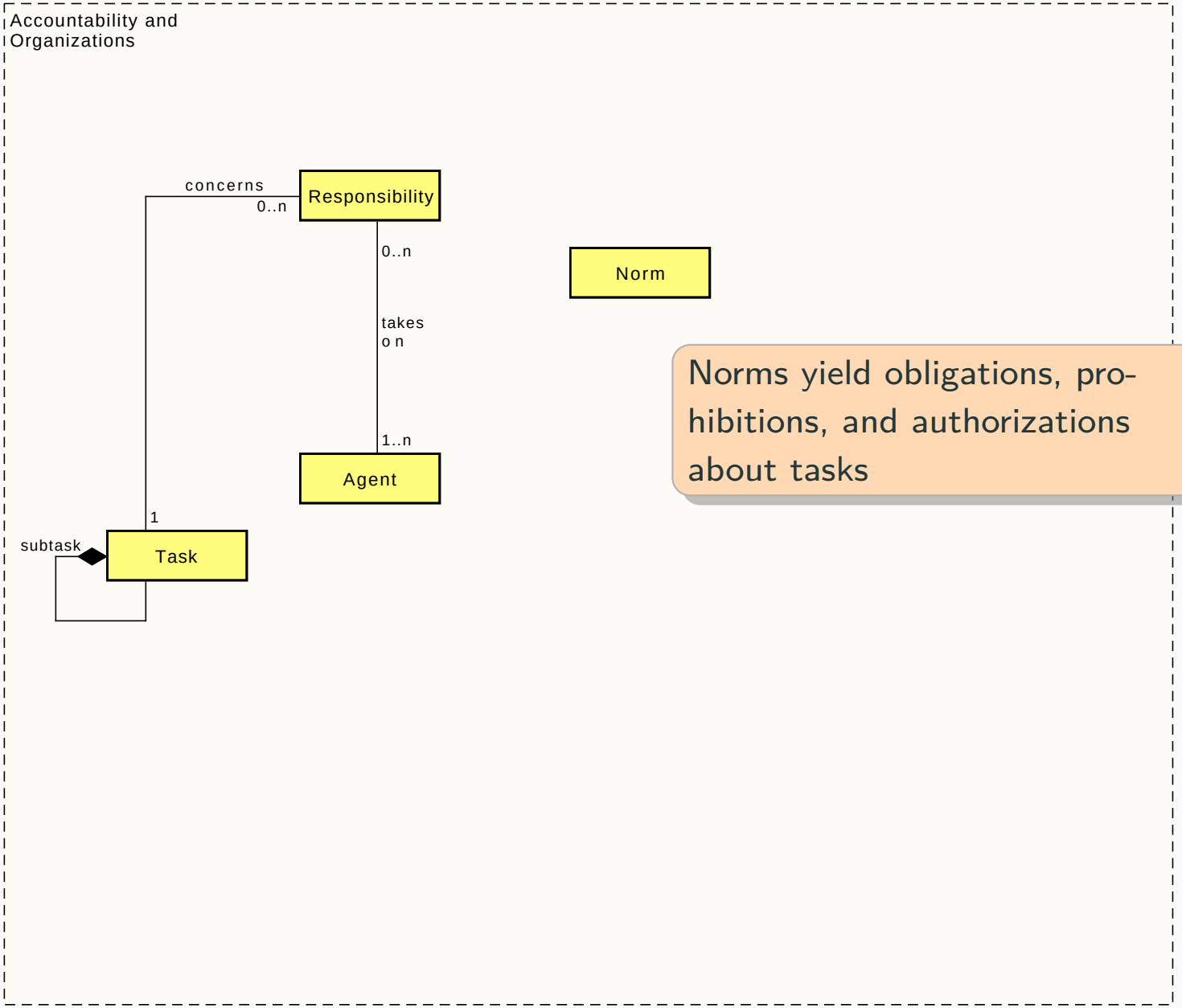
Task, Agent, and Responsibility

Accountability and Organizations



Tasks can be complex and involves sub-tasks

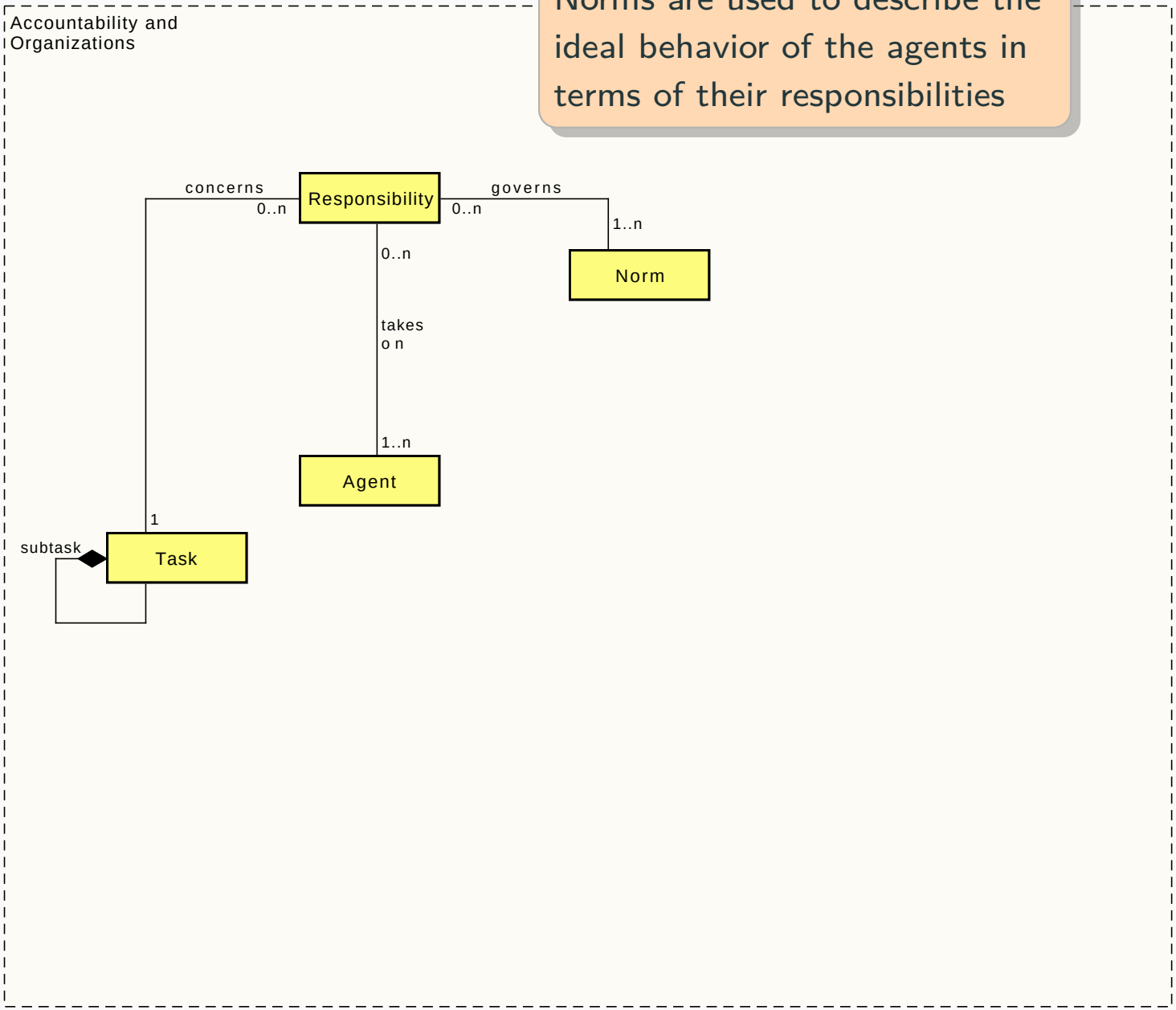
Norm



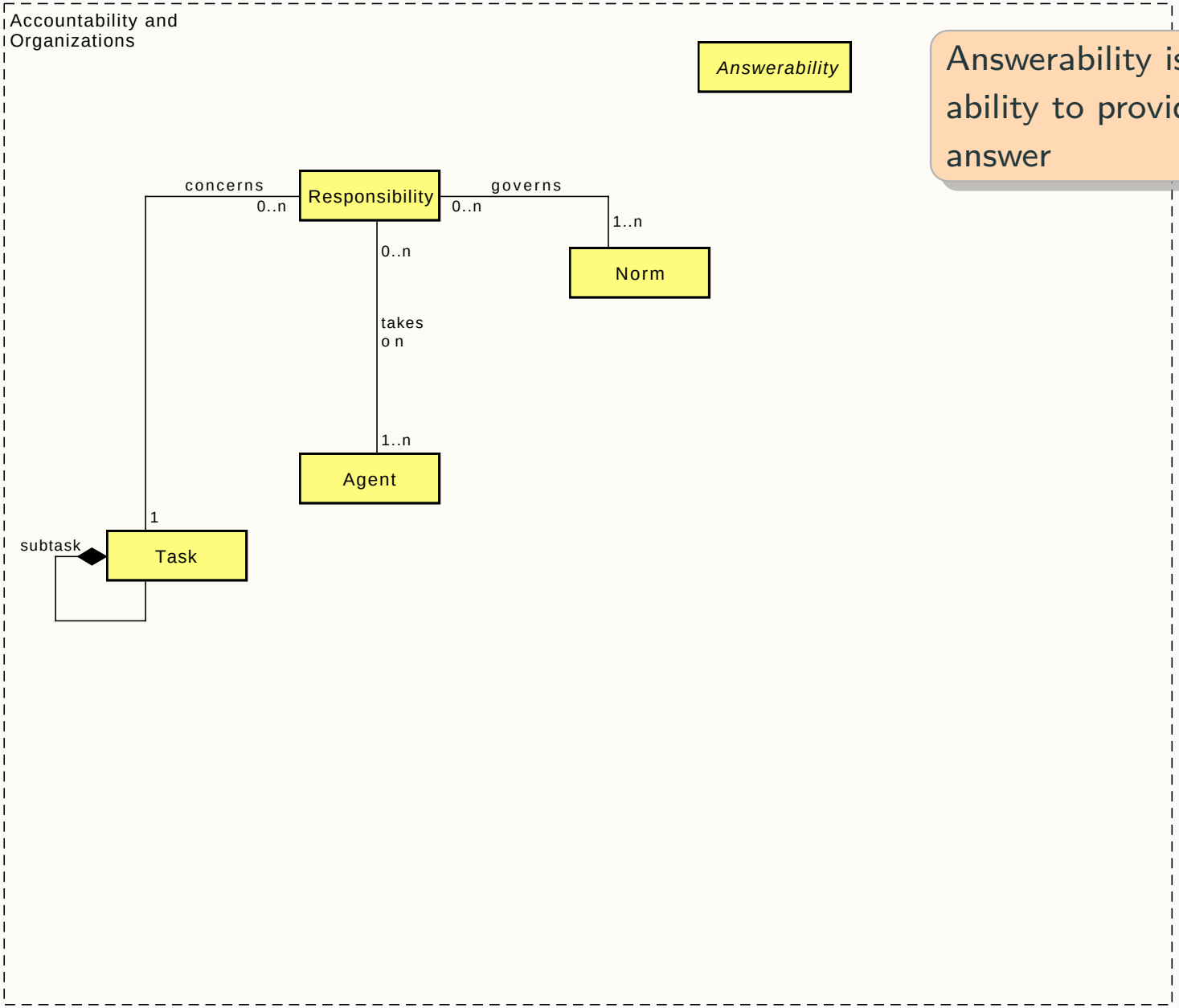
Norms yield obligations, prohibitions, and authorizations about tasks

Norm

Norms are used to describe the ideal behavior of the agents in terms of their responsibilities

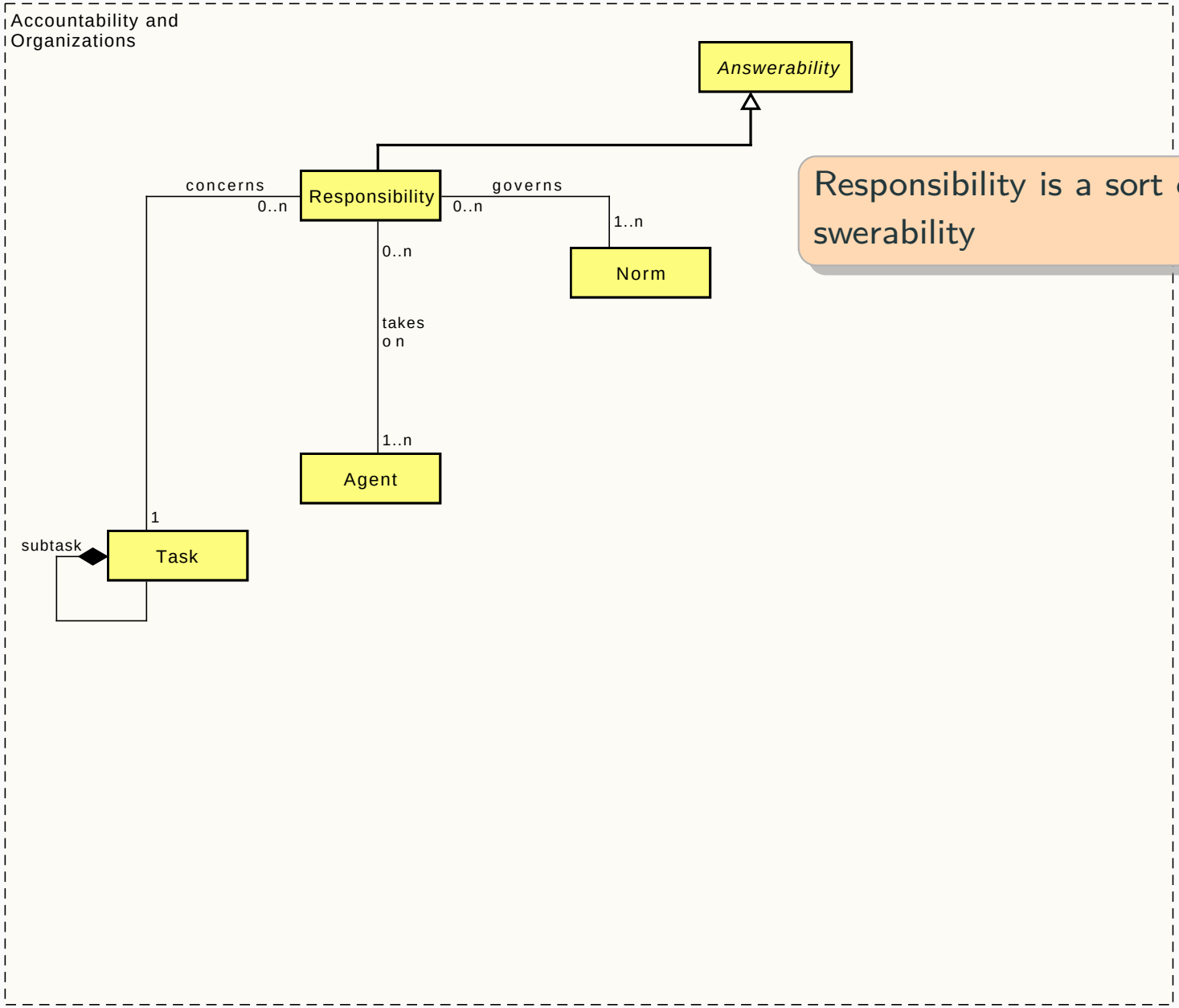


Answerability and Accountability



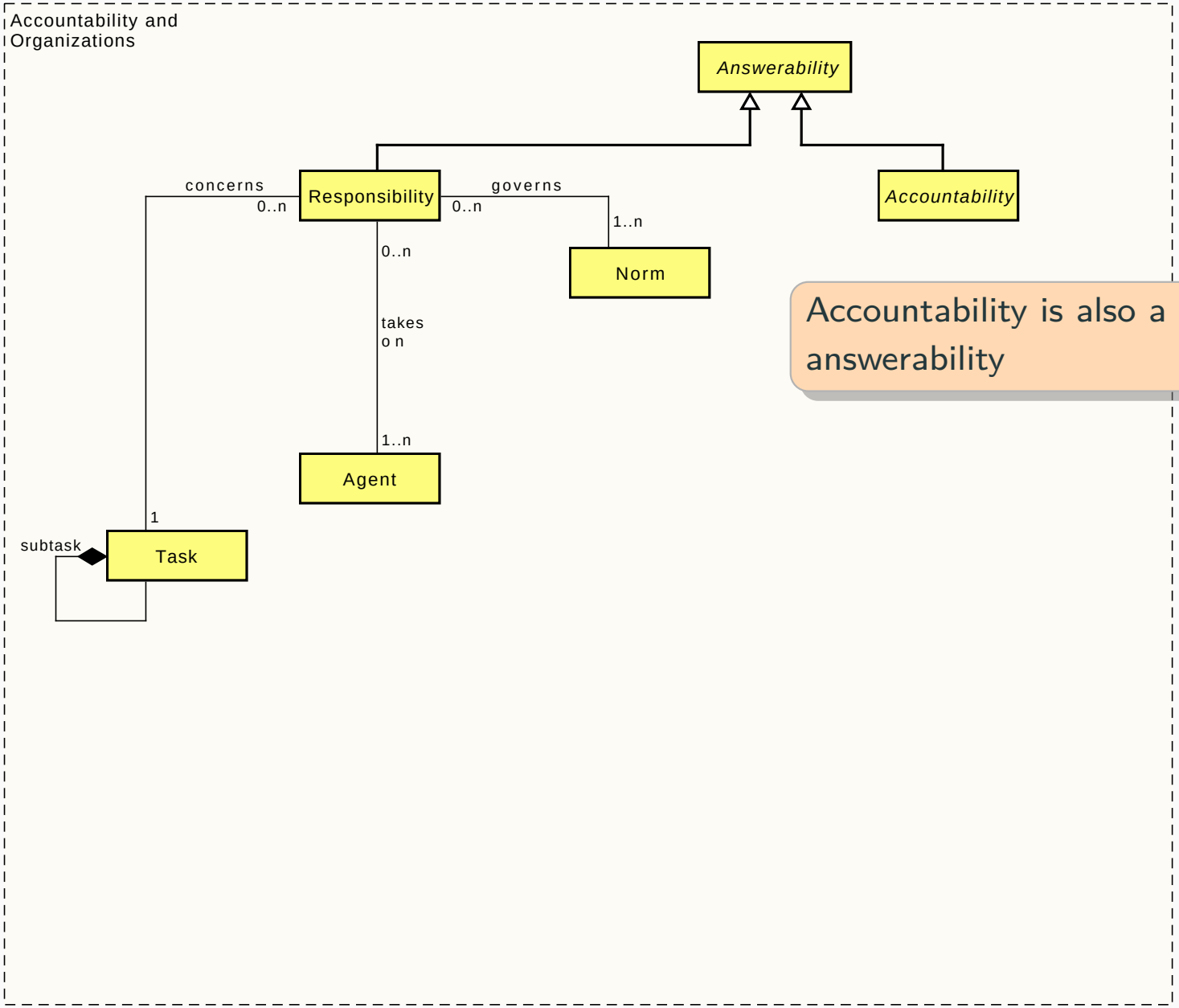
Answerability is the ability to provide an answer

Answerability and Accountability



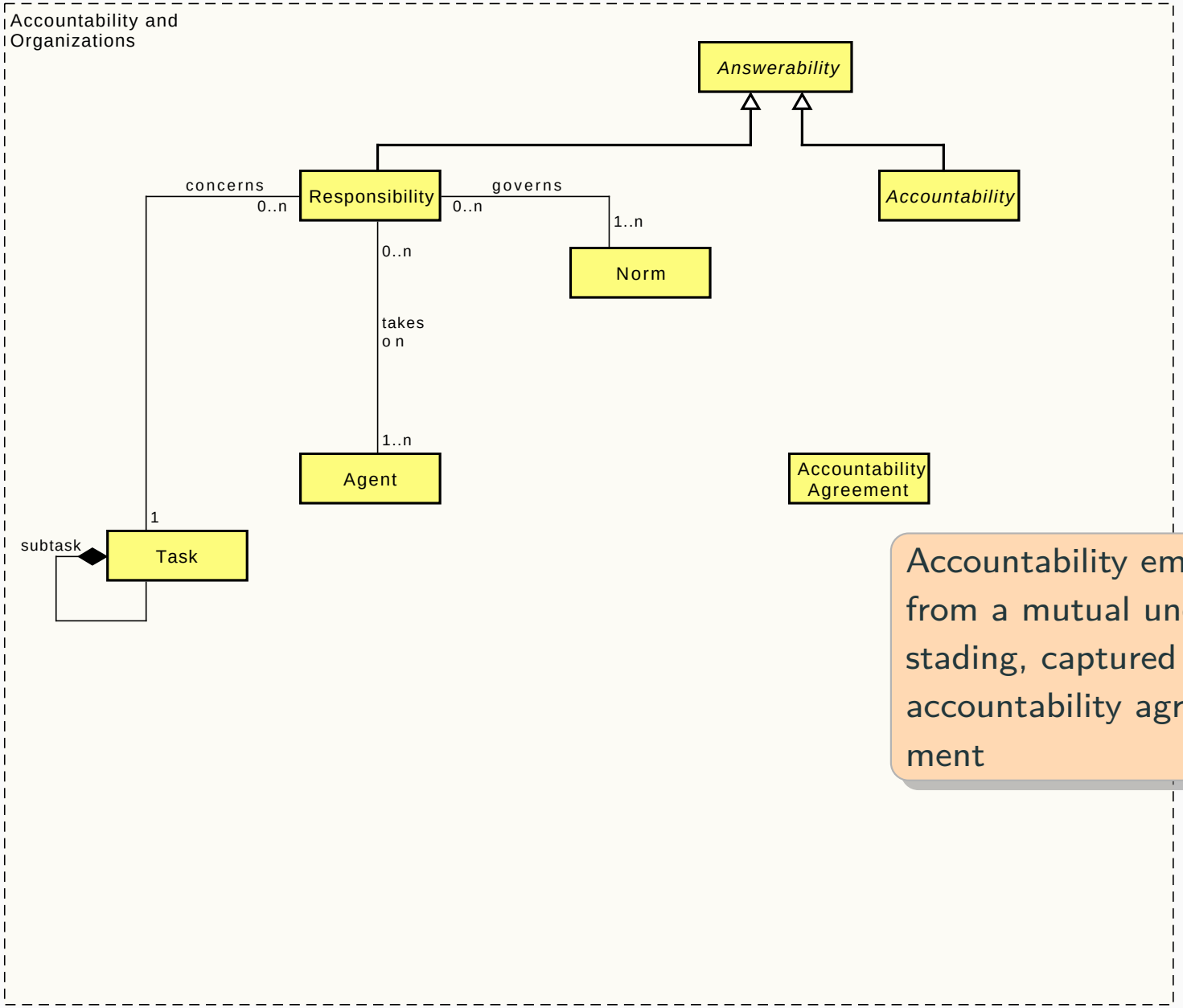
Responsibility is a sort of answerability

Answerability and Accountability



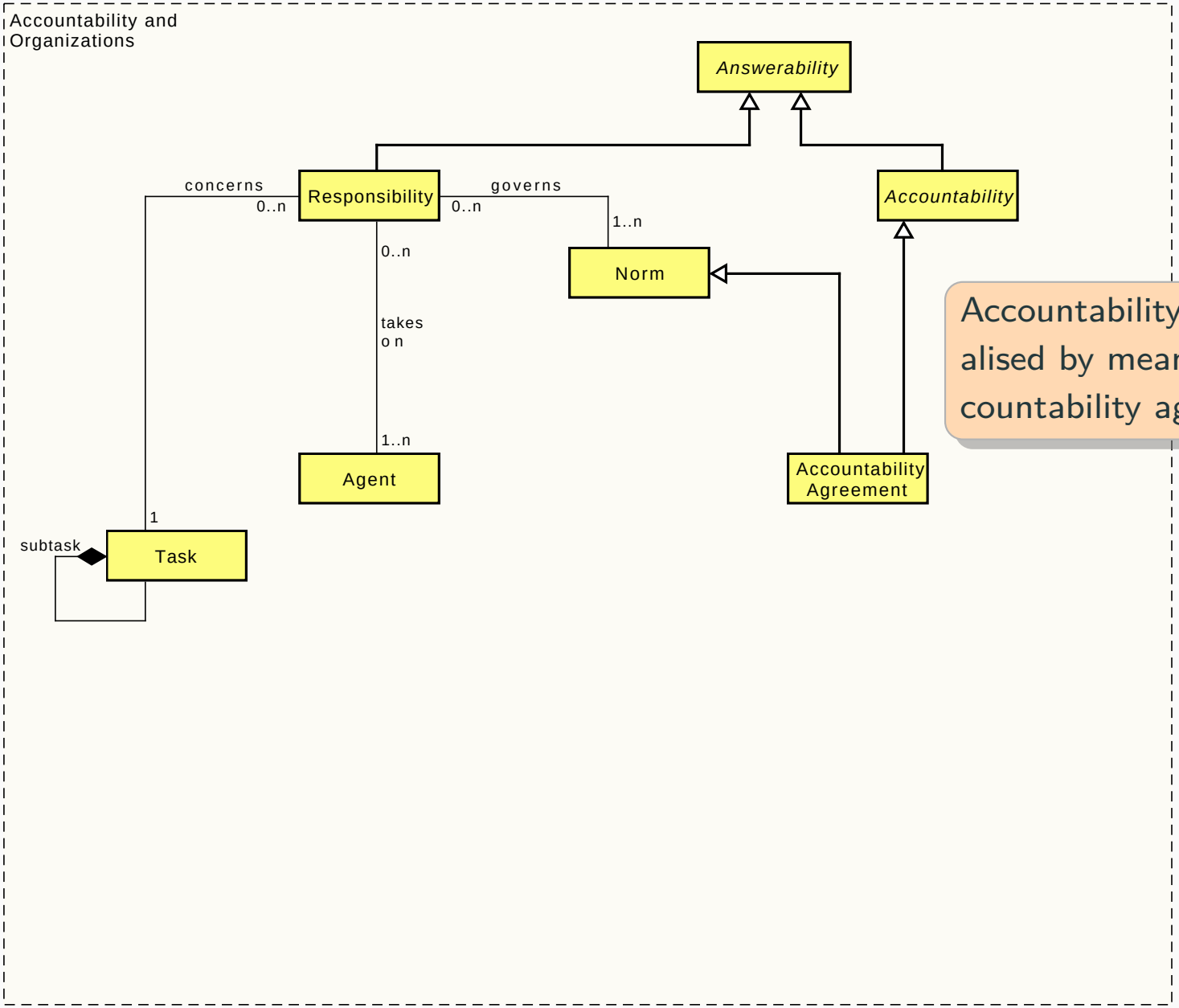
Accountability is also a sort of answerability

Accountability agreement



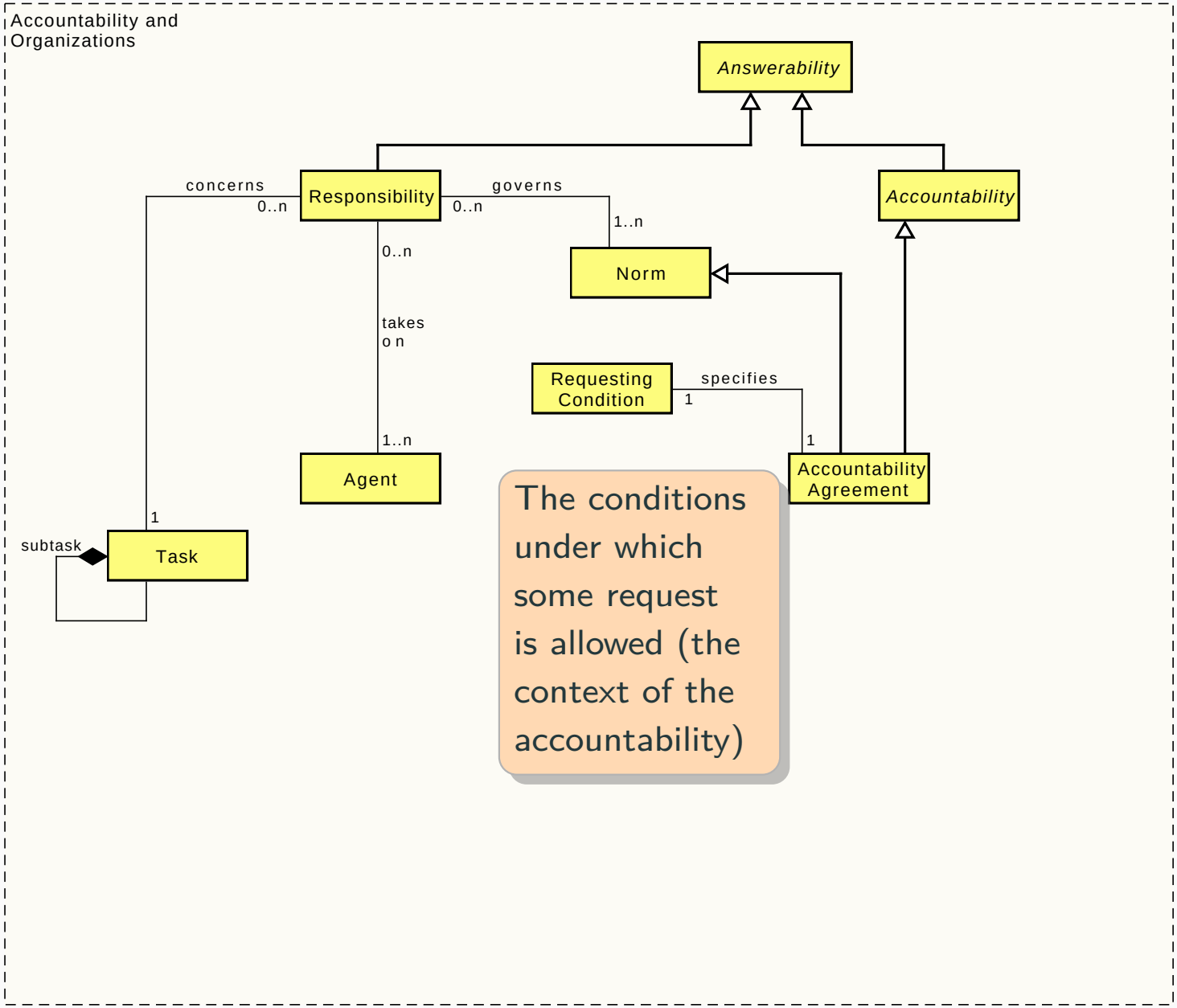
Accountability emerges from a mutual understading, captured by accountability agreement

Accountability agreement

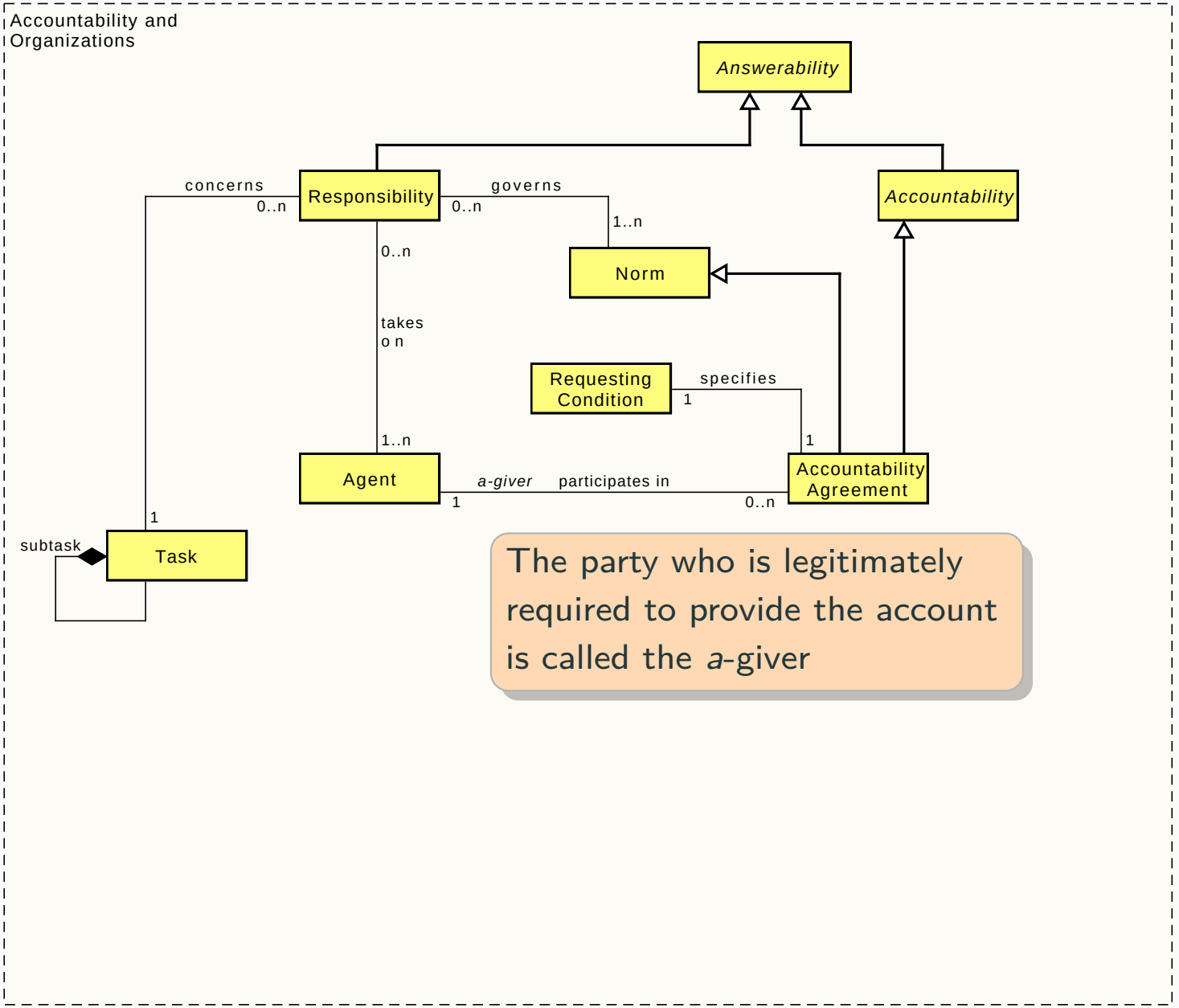


Accountability is realised by means of accountability agreements

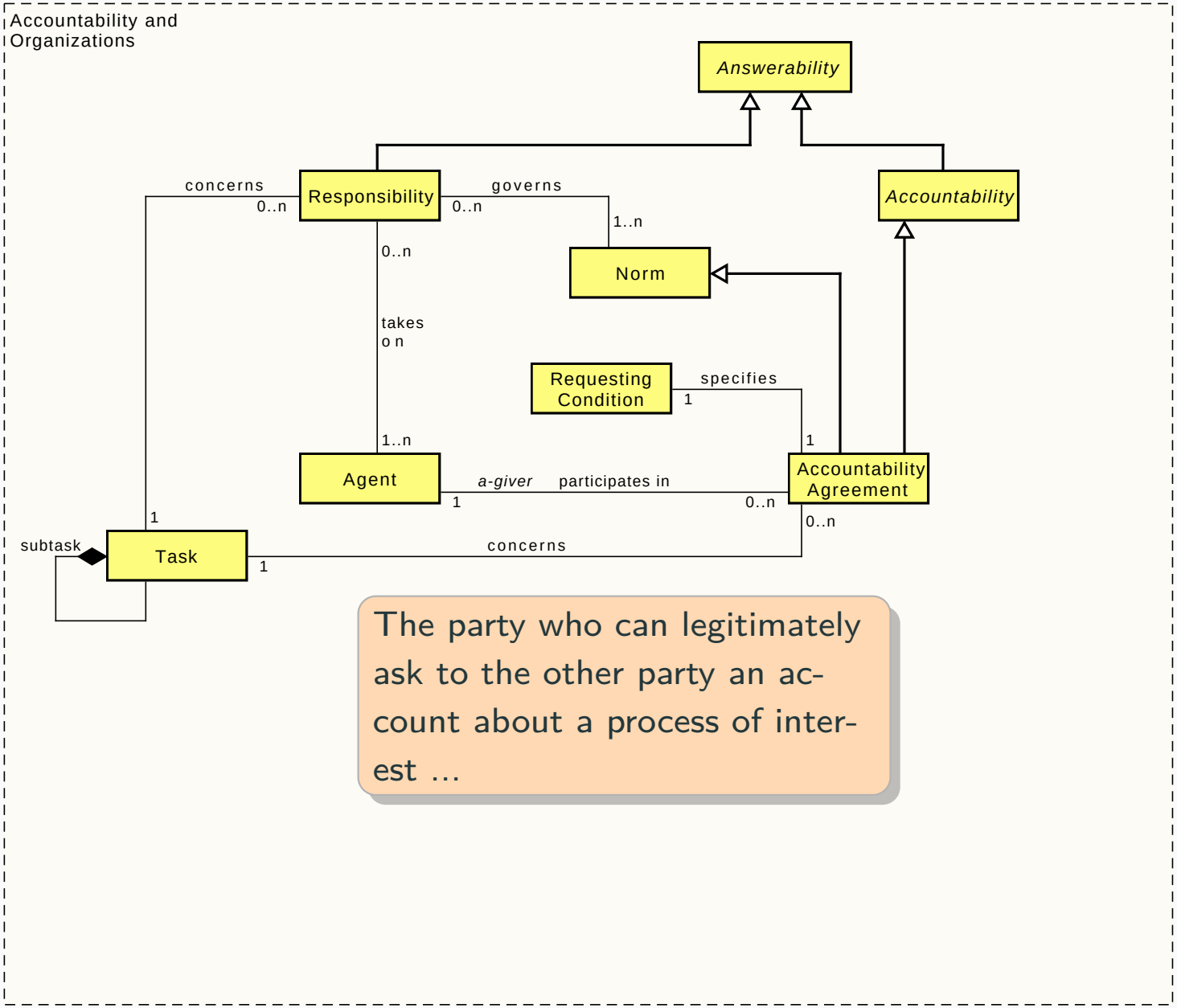
Accountability agreement



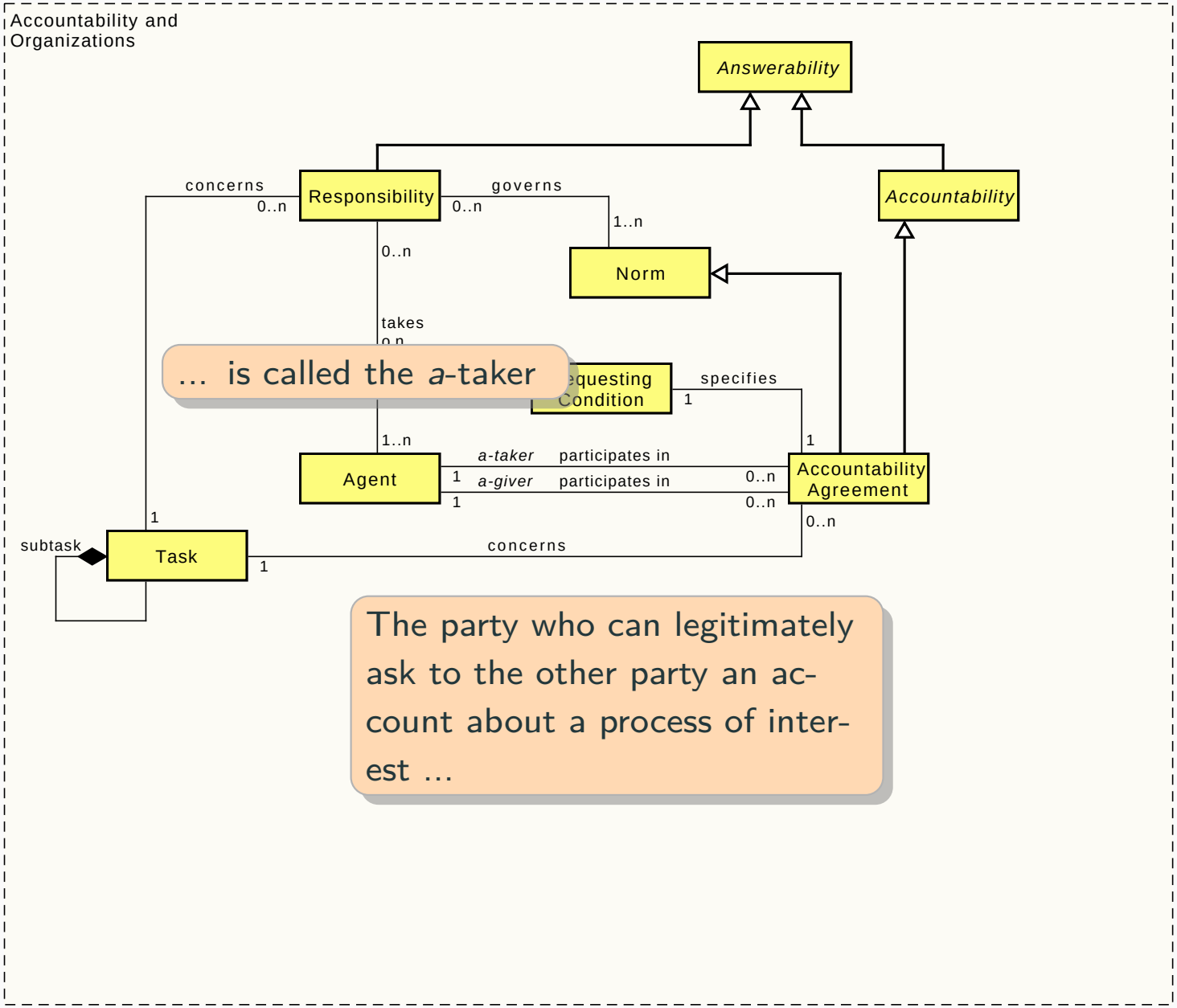
Accountability agreement



Accountability agreement



Accountability agreement



How to model the two dimension of the accountability?

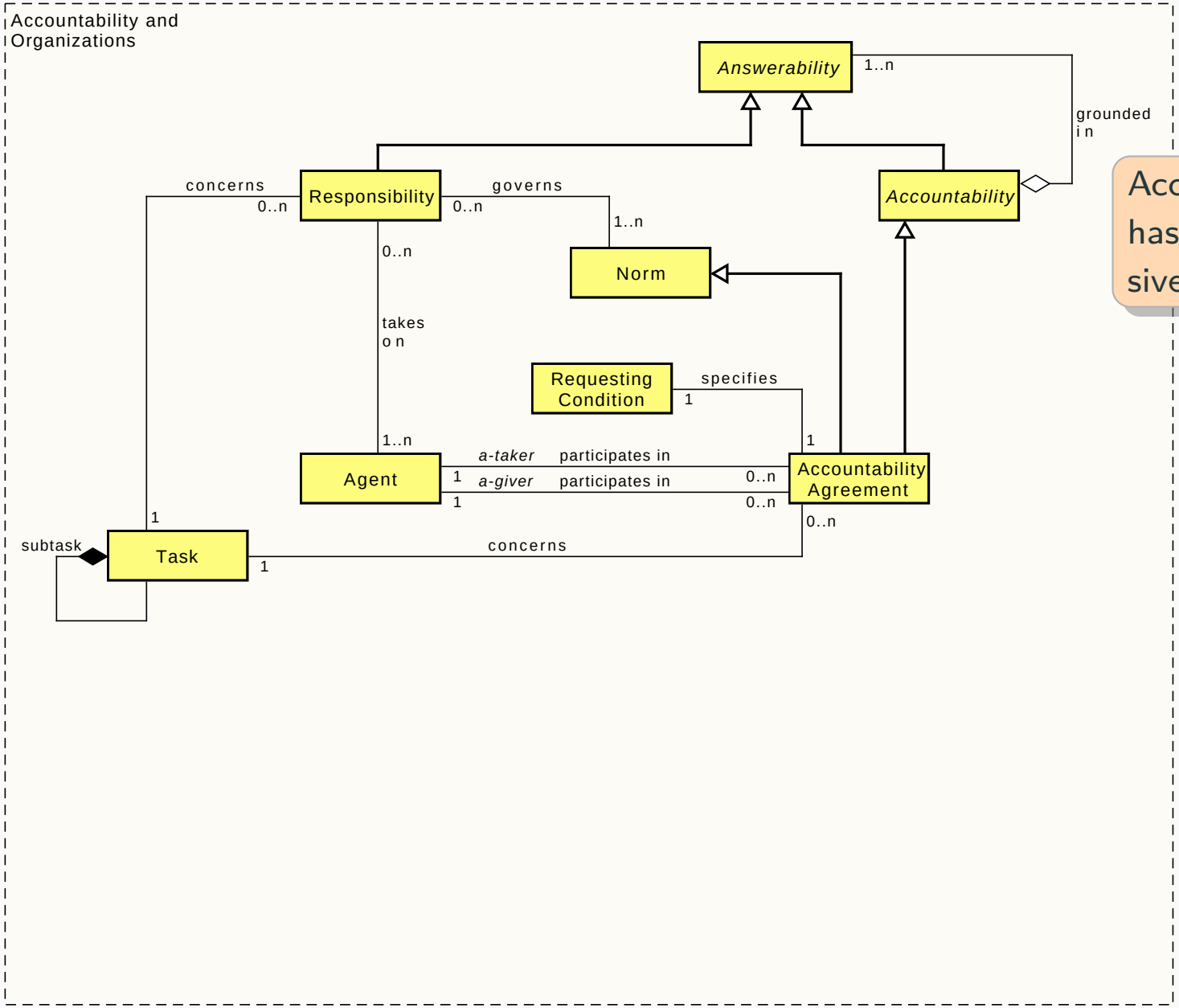
A relationship between two parties:

- One of the parties (the "**account taker**" or *a-taker*) can legitimately ask, under some agreed conditions, to the other party an **account** about a process of interest
- the other party (the "**account giver**" or *a-giver*) is legitimately required to provide the account to the a-taker

The two dimensions of accountability

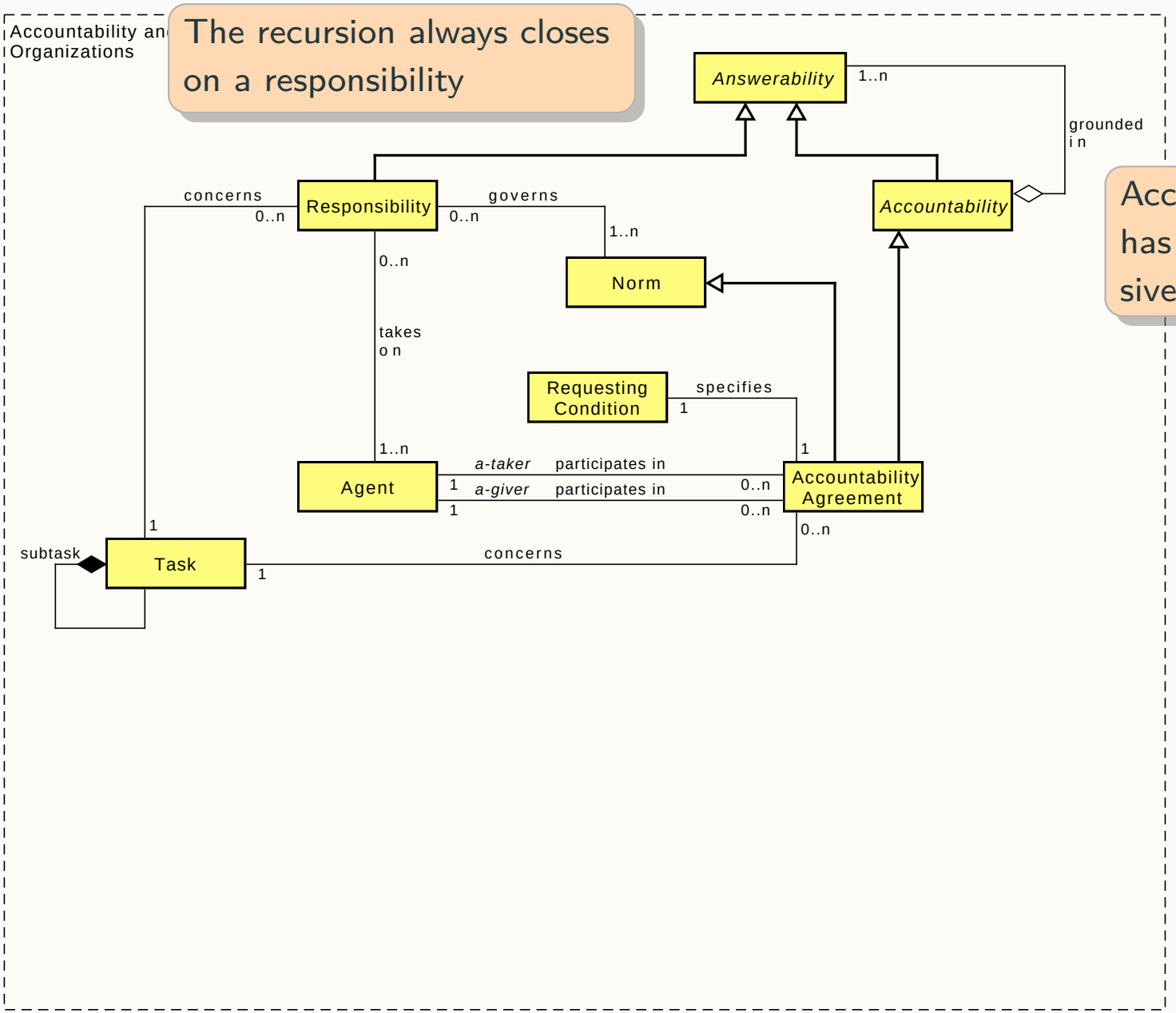
1. **Normative dimension** $\rightarrow$ Legitimacy of asking and availability to provide accounts
2. **Structural dimension** $\rightarrow$ For being accountable about a process, an agent must have control over that process and have awareness of the situation it will account for

Accountability: the decorator pattern



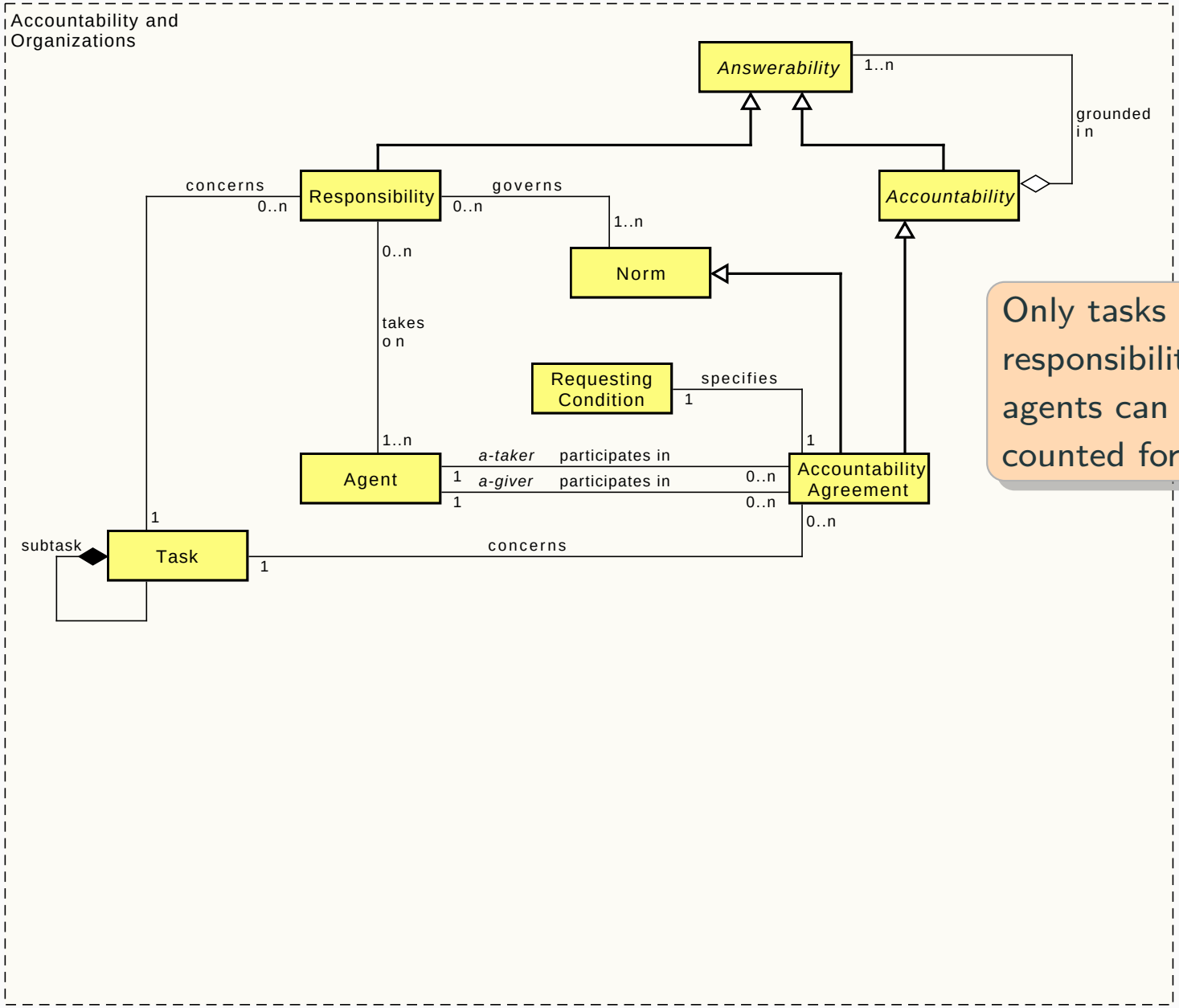
Accountability has a recursive structure

Accountability: the decorator pattern



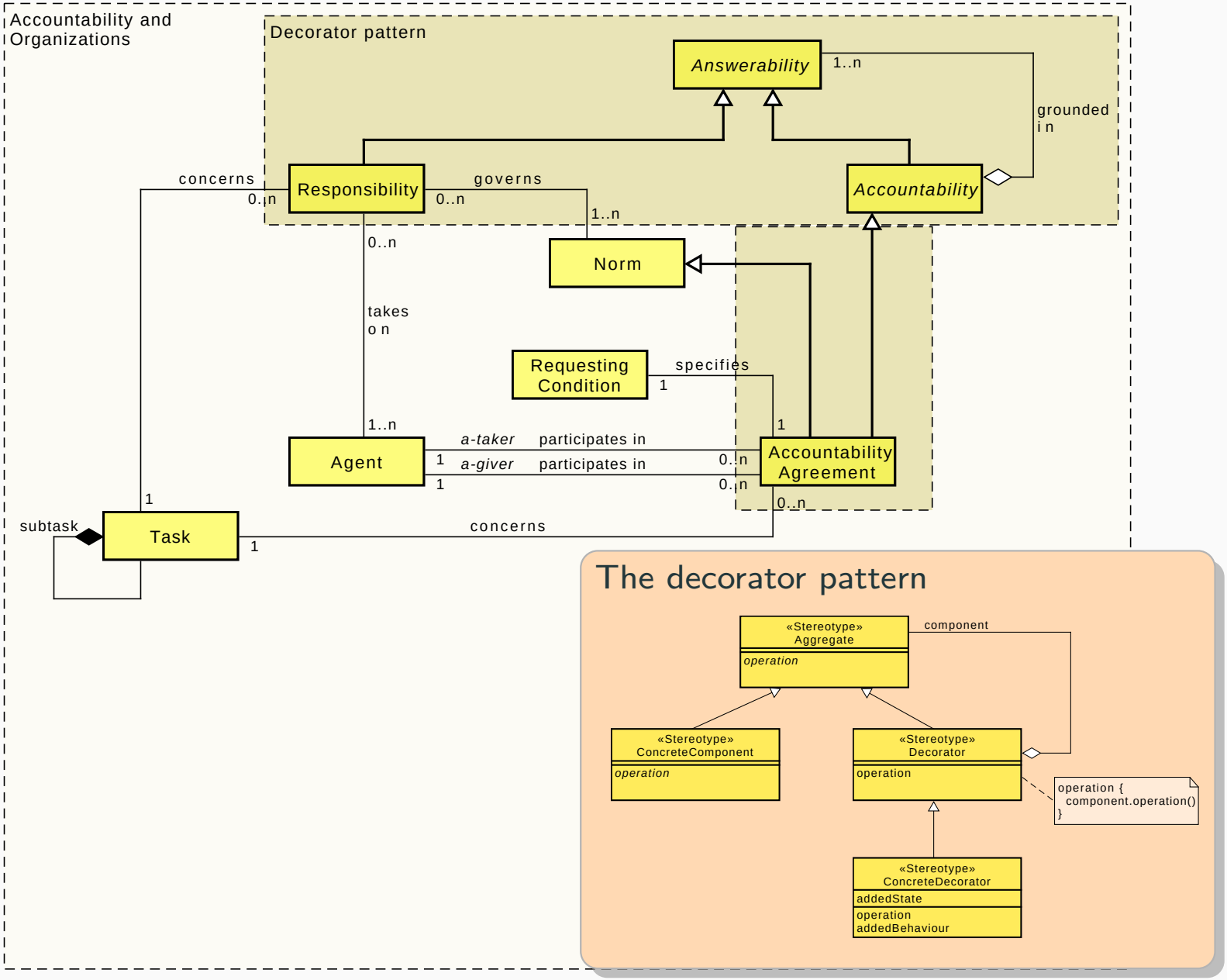
Accountability has a recursive structure

Accountability: the decorator pattern

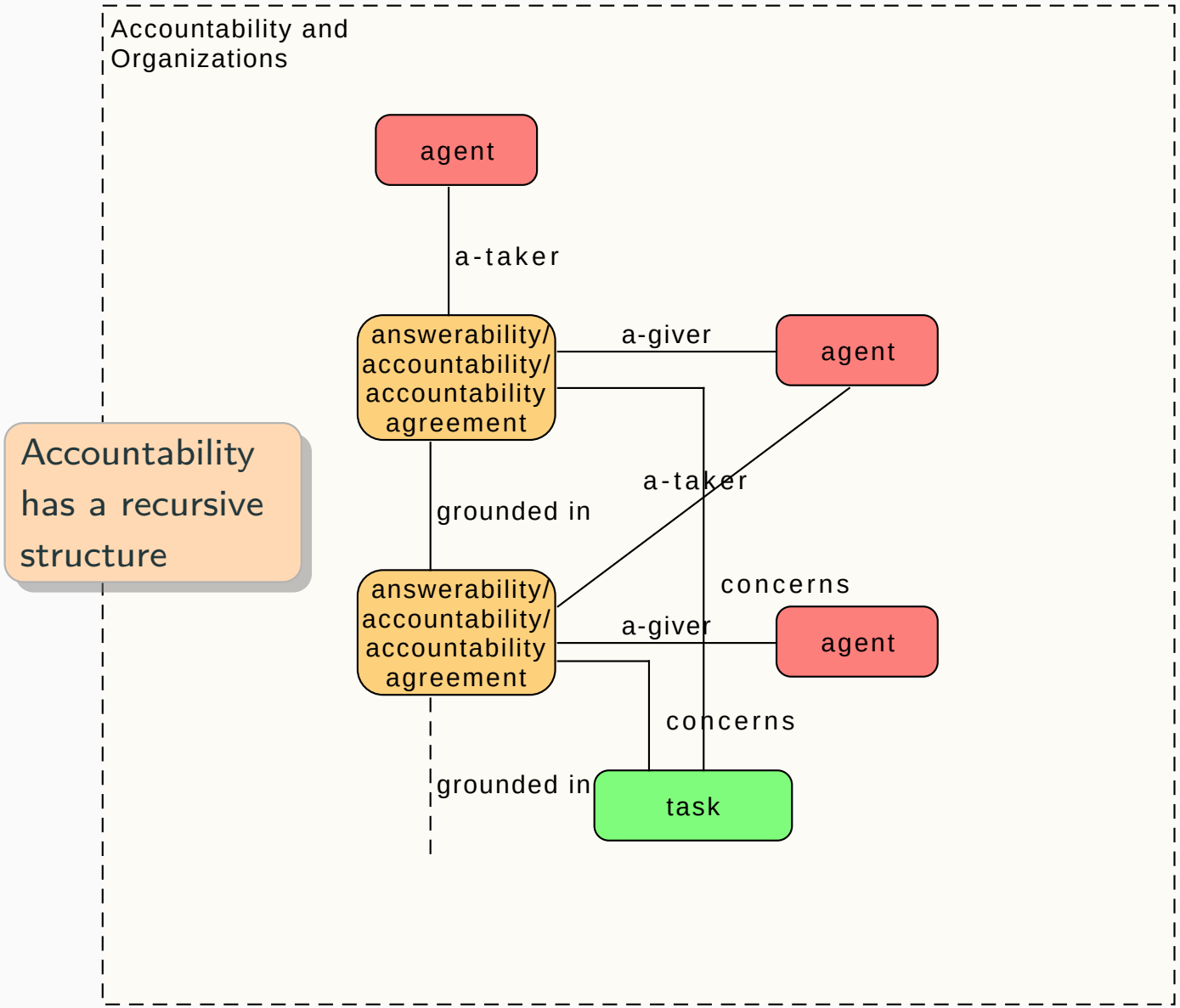


Only tasks under the responsibility of some agents can be accounted for

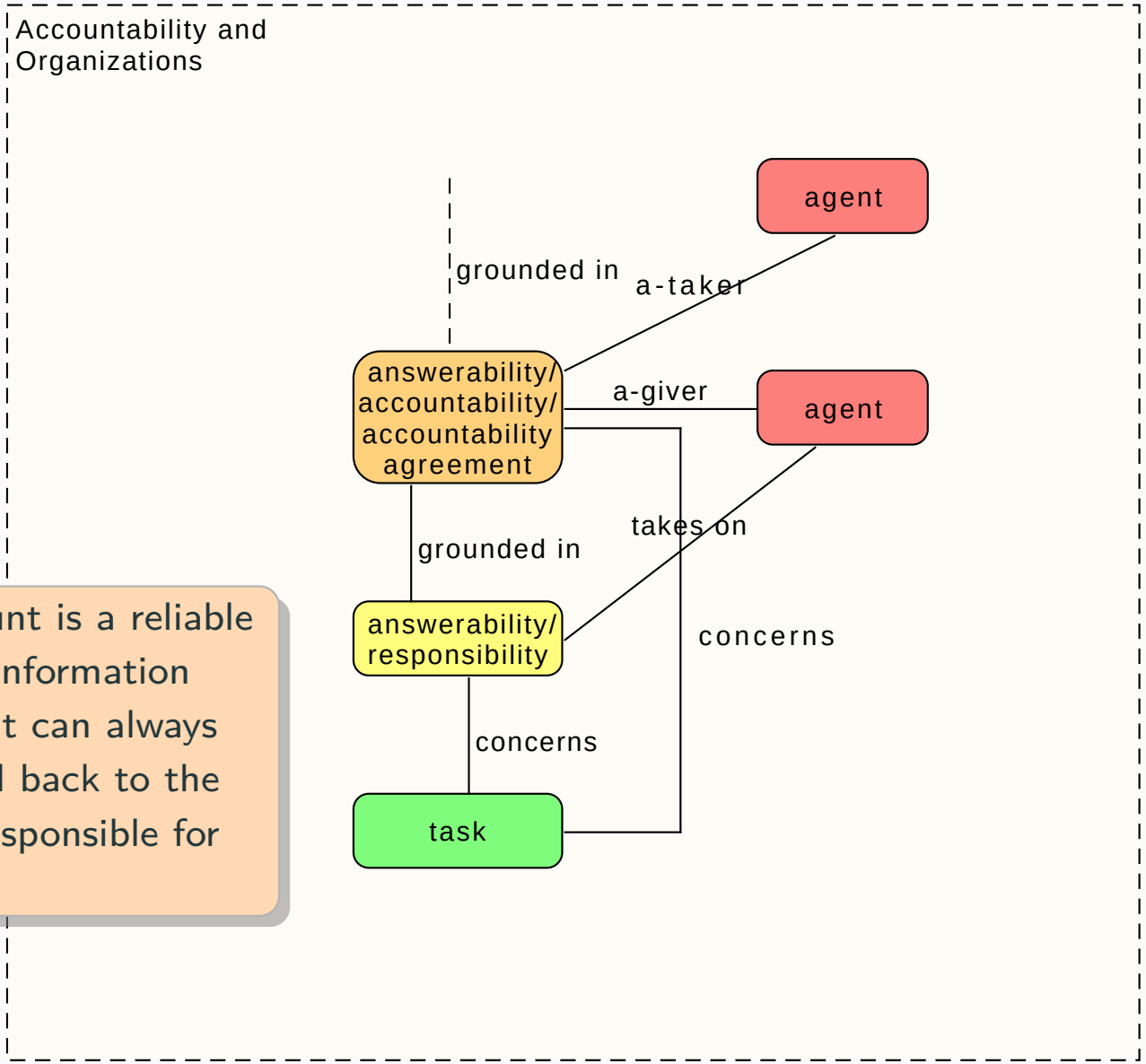
Accountability: the decorator pattern



From the point of view of instances

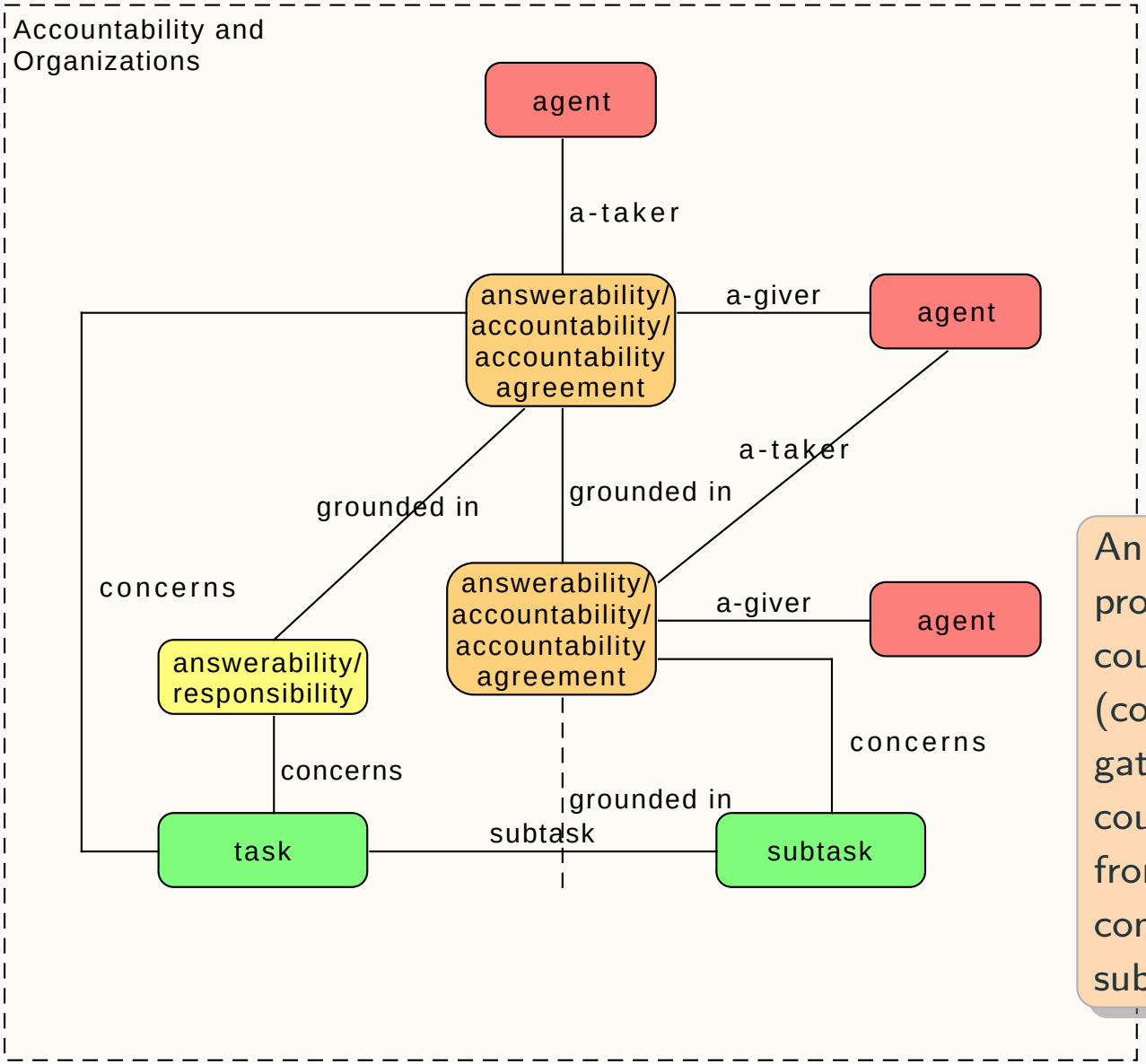


From the point of view of instances



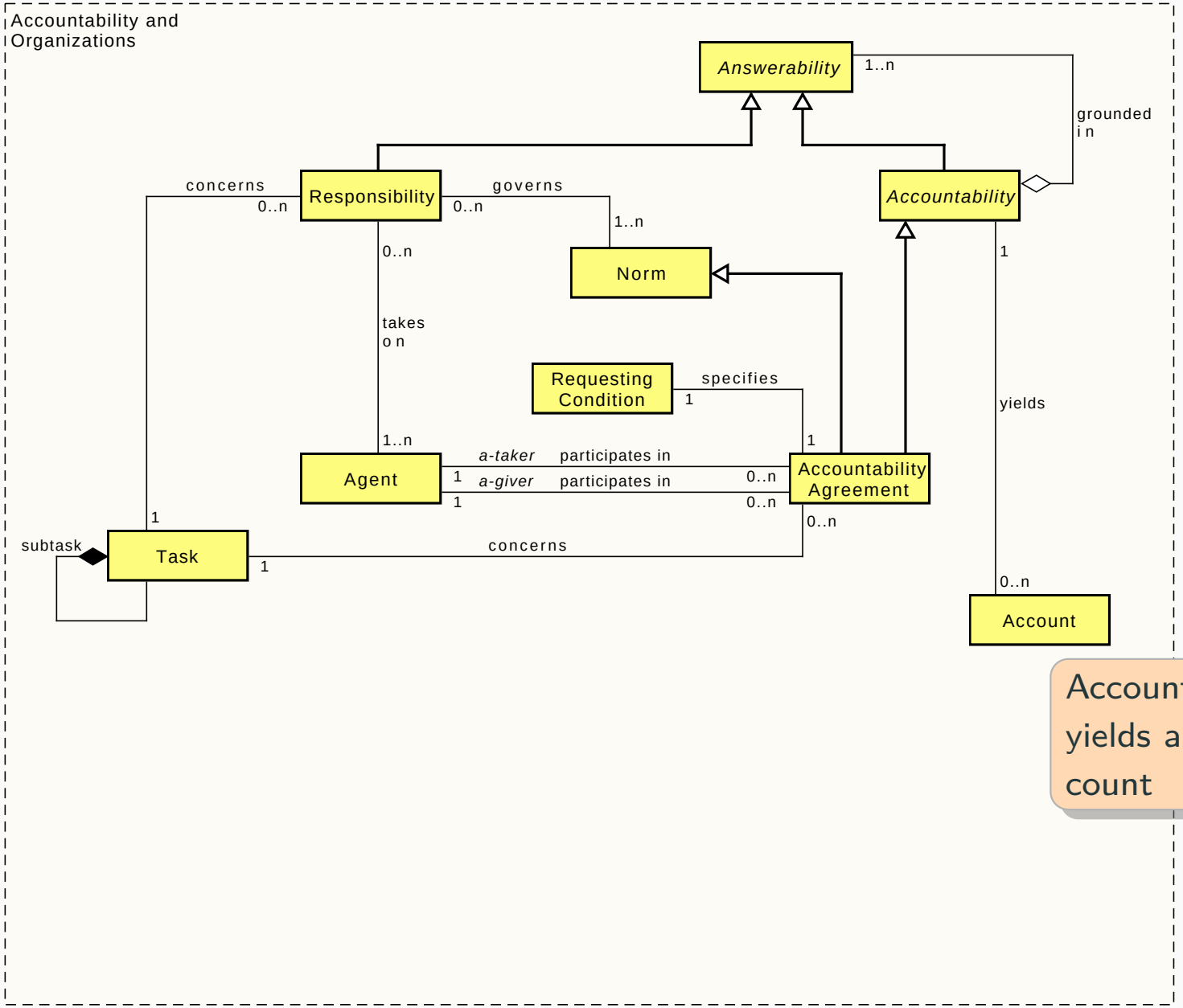
An account is a reliable piece of information because it can always be traced back to the agents responsible for that task

From the point of view of instances



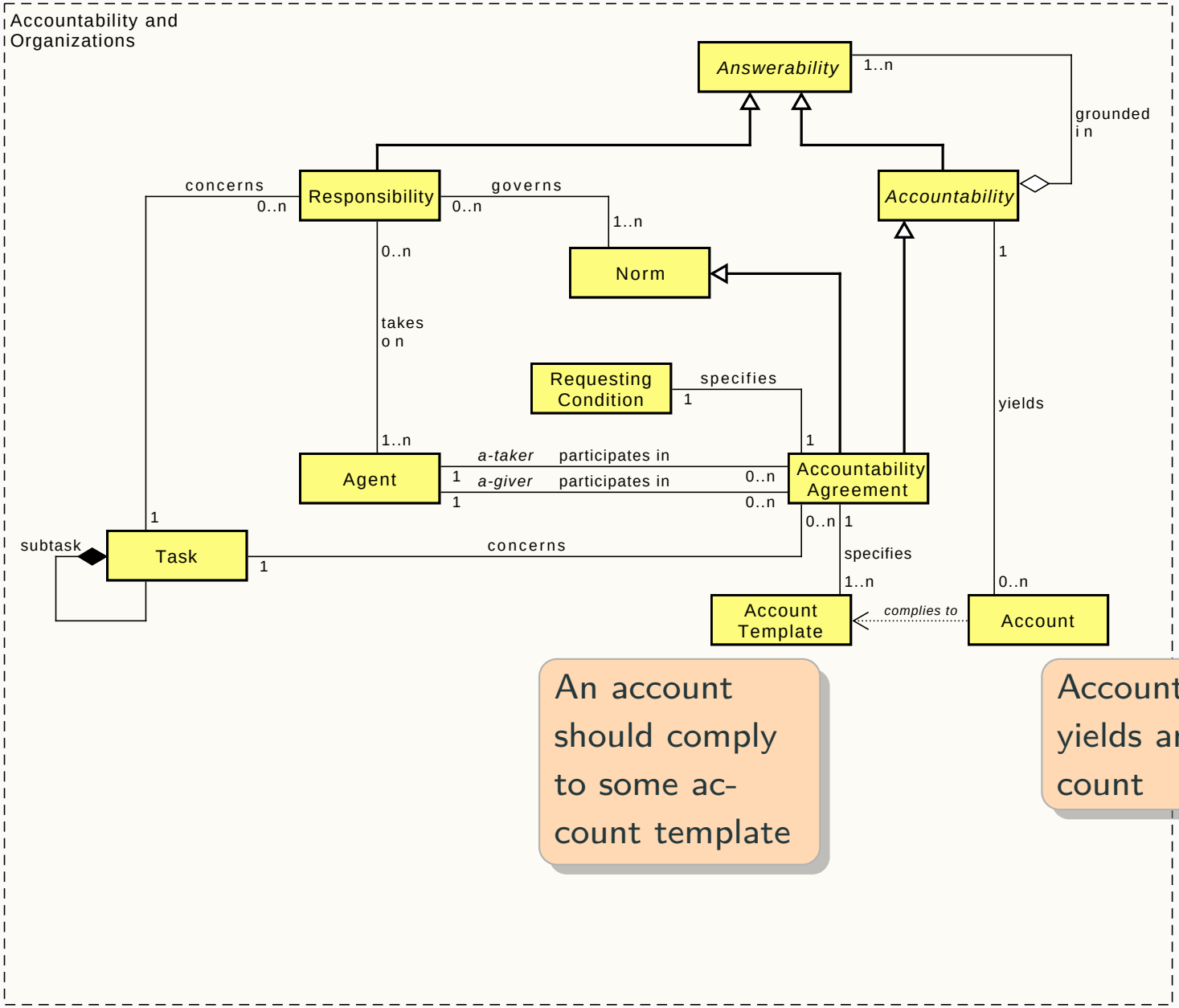
An agent can provide an account about a (complex) task by gathering the accounts it receives from other agents concerning other subtasks

Account and Account Template

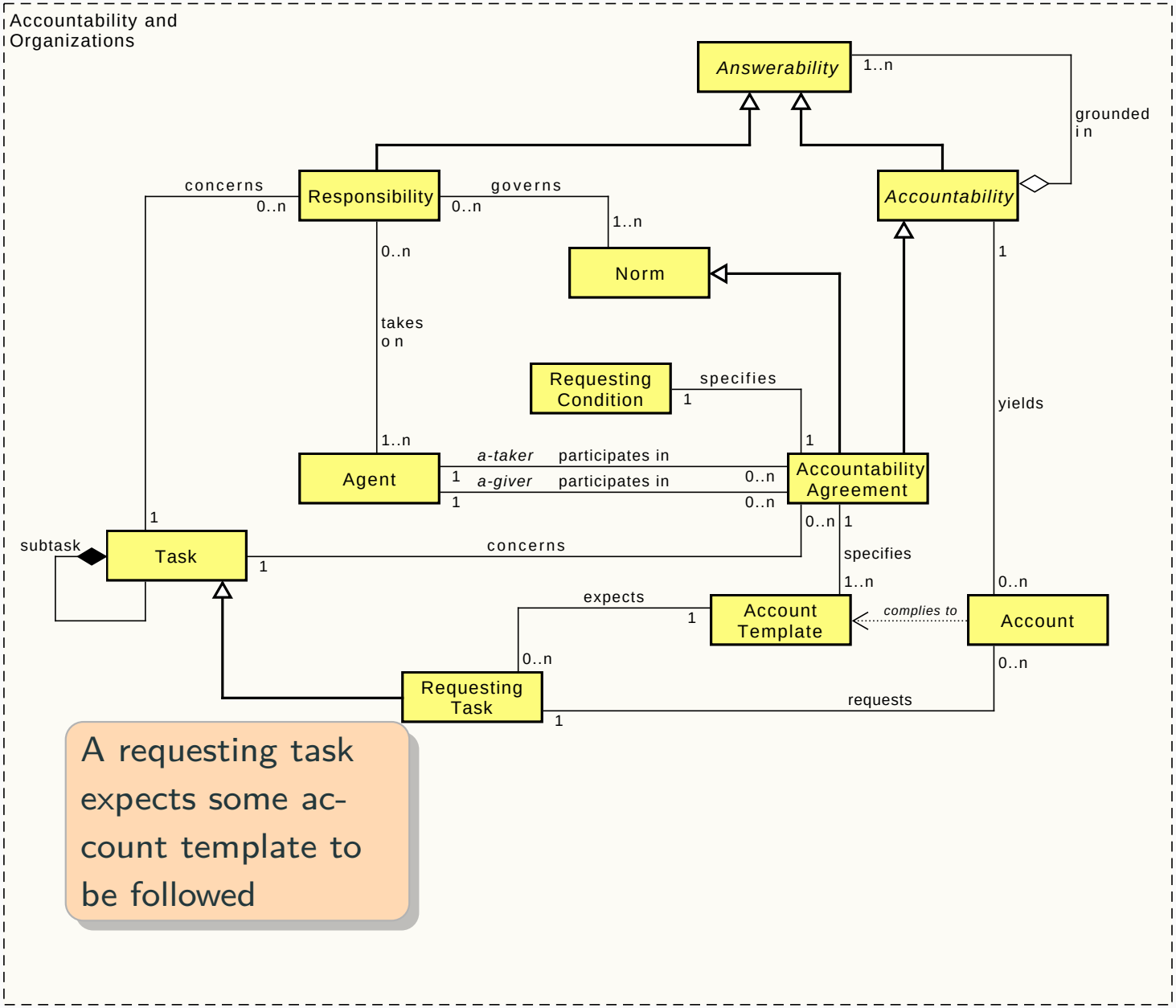


Accountability yields an account

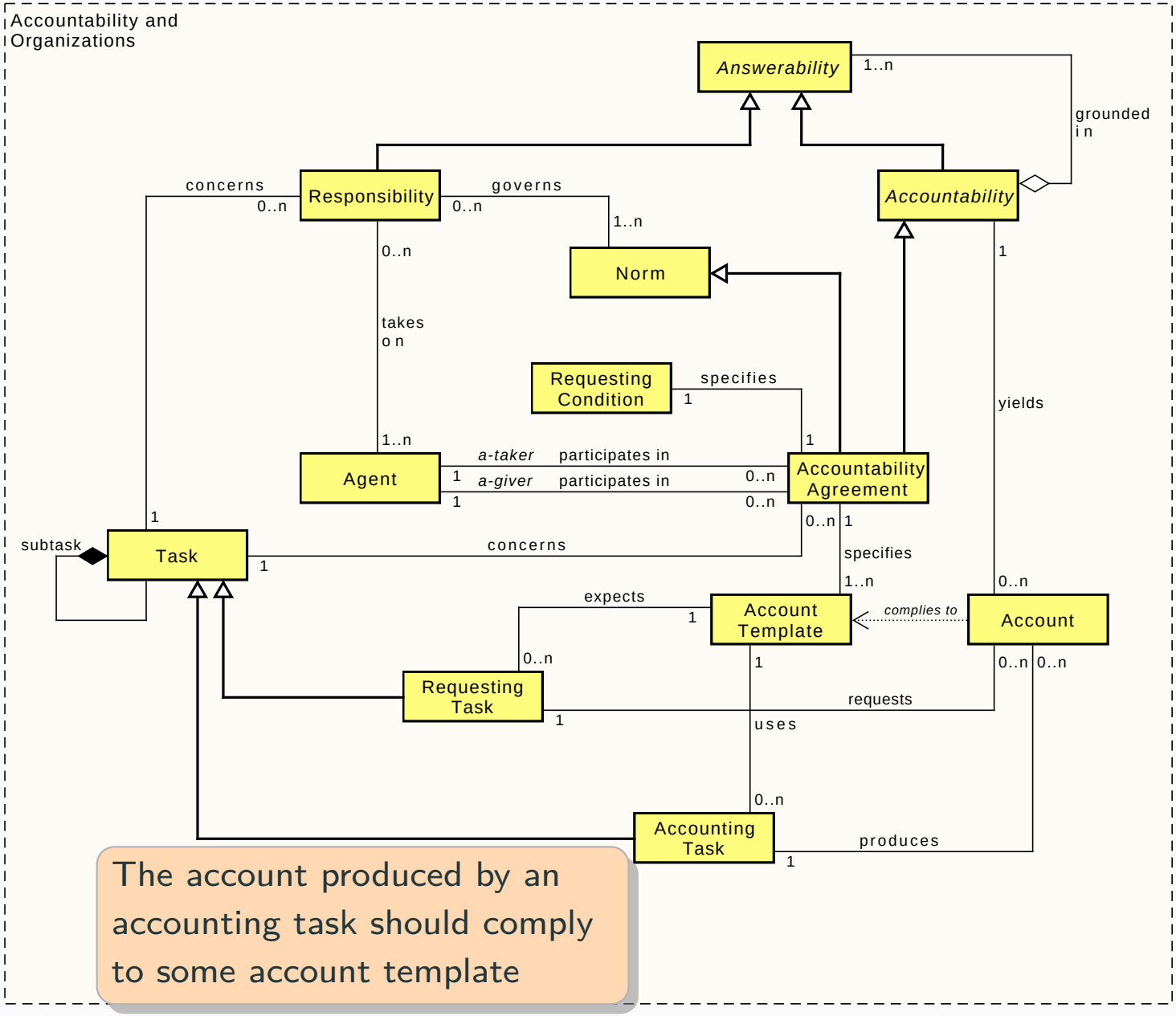
Account and Account Template



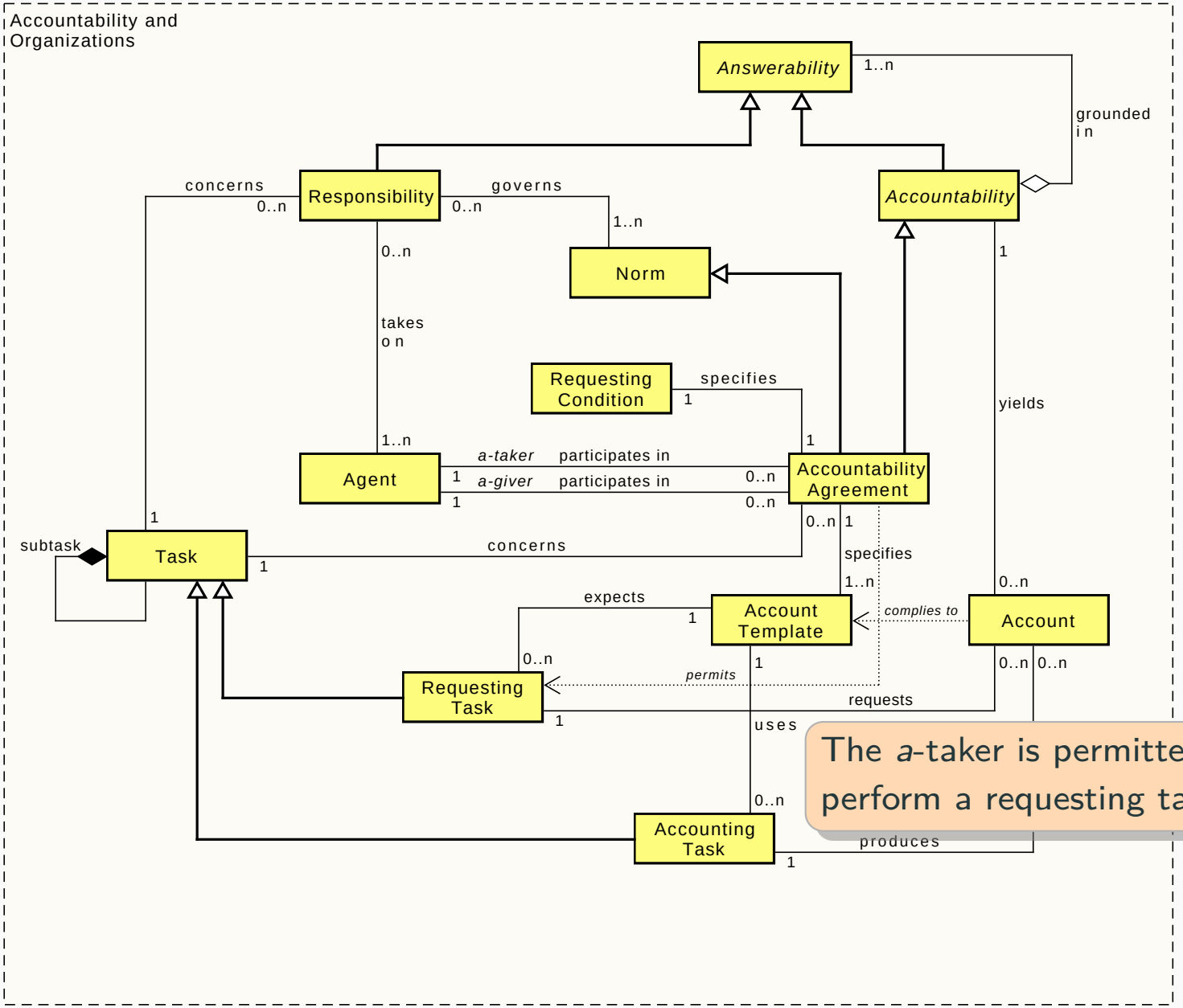
Requesting and Accounting Task



Requesting and Accounting Task

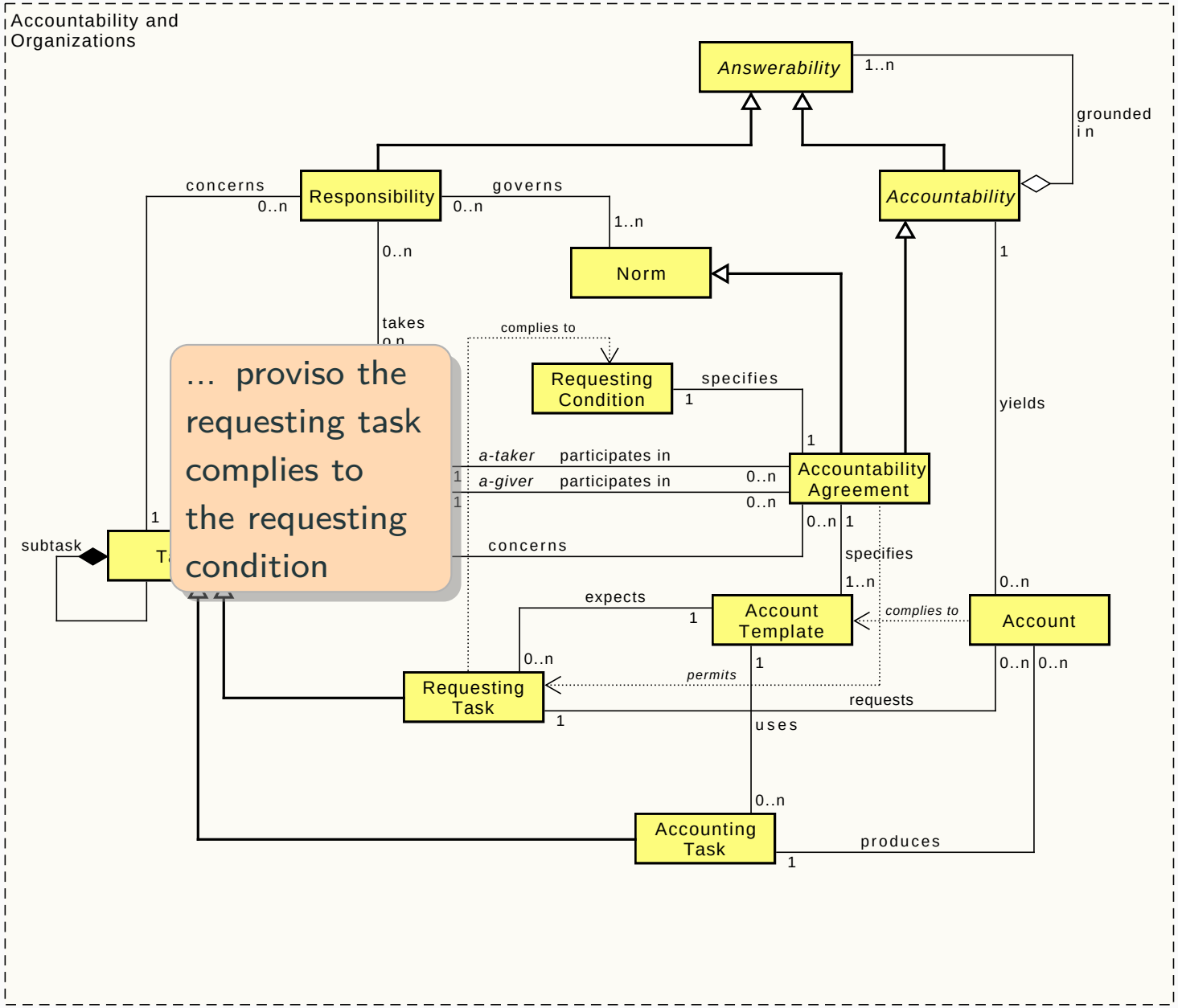


Requesting and Accounting Task

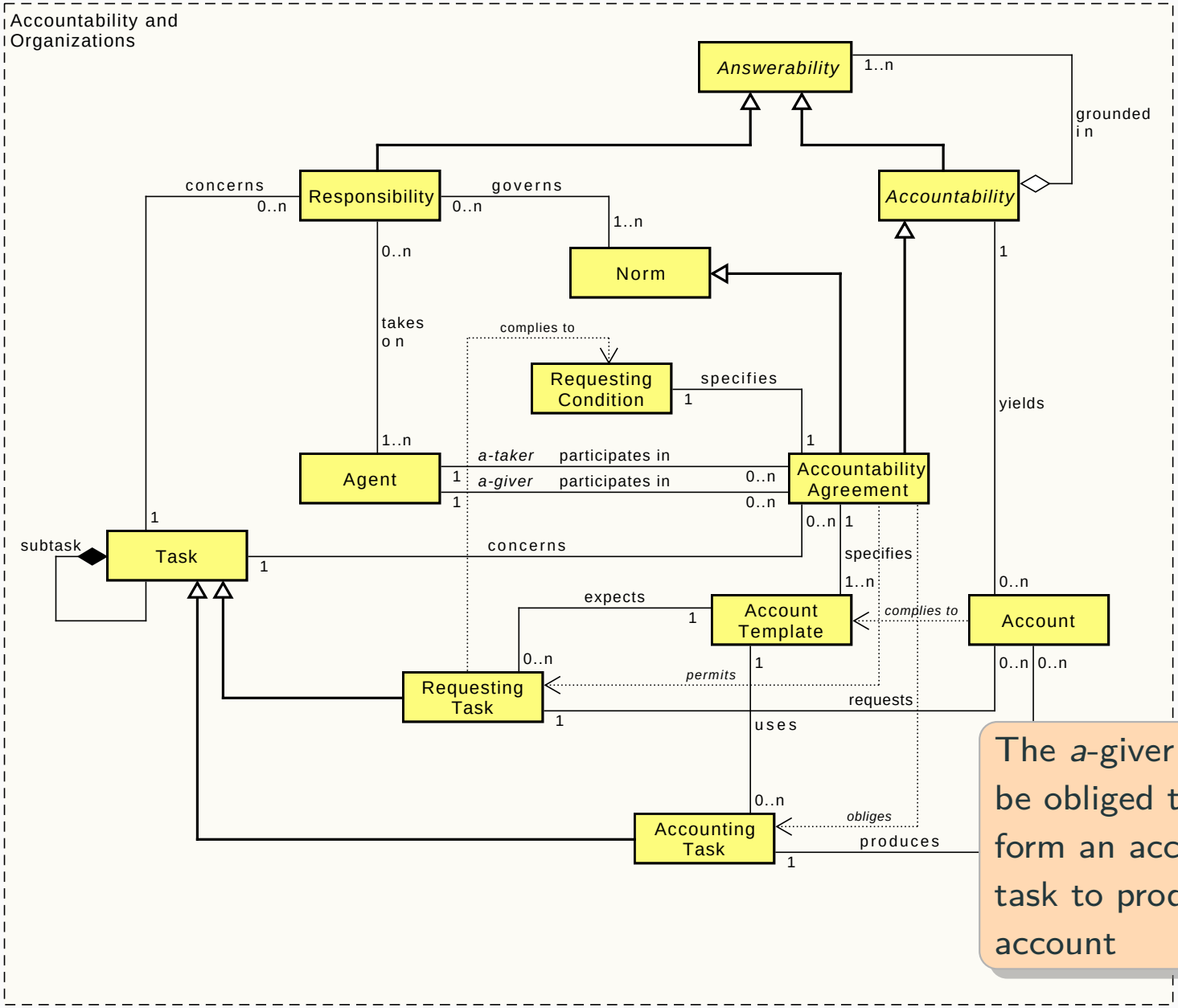


The *a*-taker is permitted to perform a requesting task

Requesting and Accounting Task

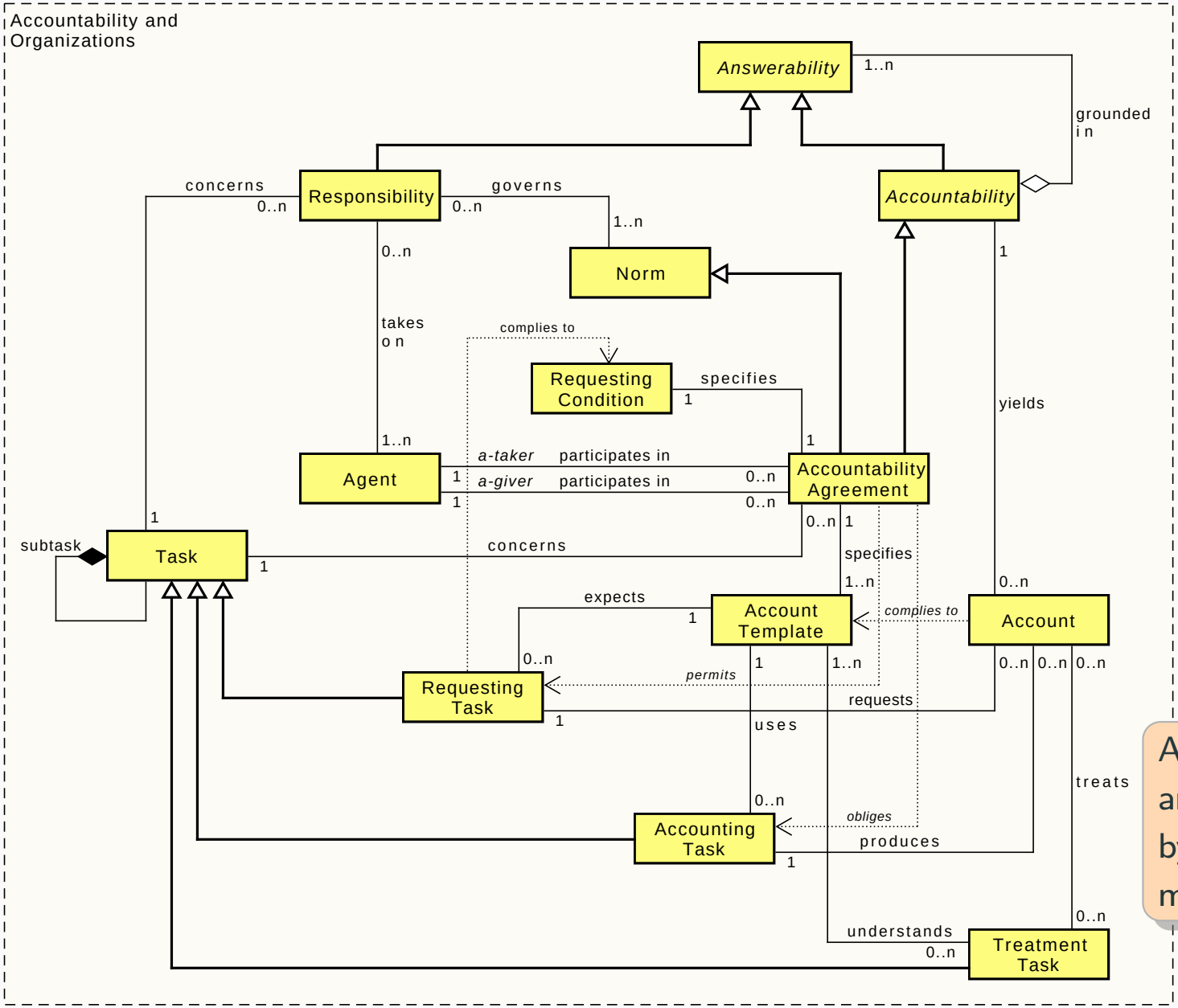


Requesting and Accounting Task



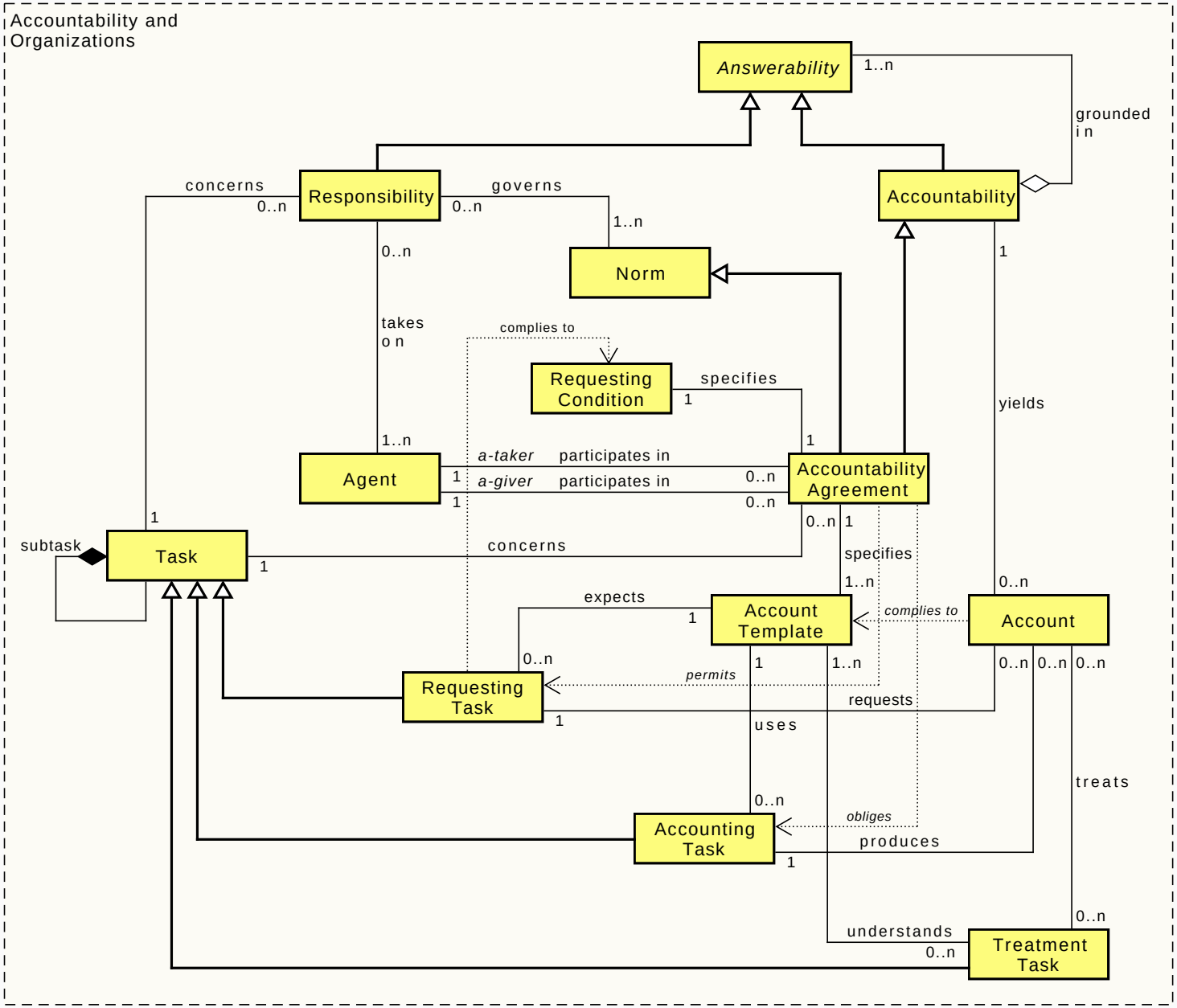
The *a*-giver may be obliged to perform an accounting task to produce an account

Treatment Task



Accounts are treated by treatment tasks

Organizational accountability: a conceptual model



Experimenting Accountability in JaCaMo

Outline

- A very short introduction to JaCaMo
- How JaCaMo is extended to accommodate accountability
- Patterns for using accountability in agent programming
  - Information gathering
  - Context-aware adaptation
  - Exception Handling

A very short introduction about JaCaMo

- JaCaMo is a Multi-Agent Oriented Programming (MAOP) platform
- it aims at programming systems by providing a seamless integration of three dimensions:

- Organization via **Mo**ise [Hübner et al., 2010]
- Agent via **Ja**son [Bordini et al., 2007],
- Environment via **CA**rtAgO [Ricci et al., 2009],

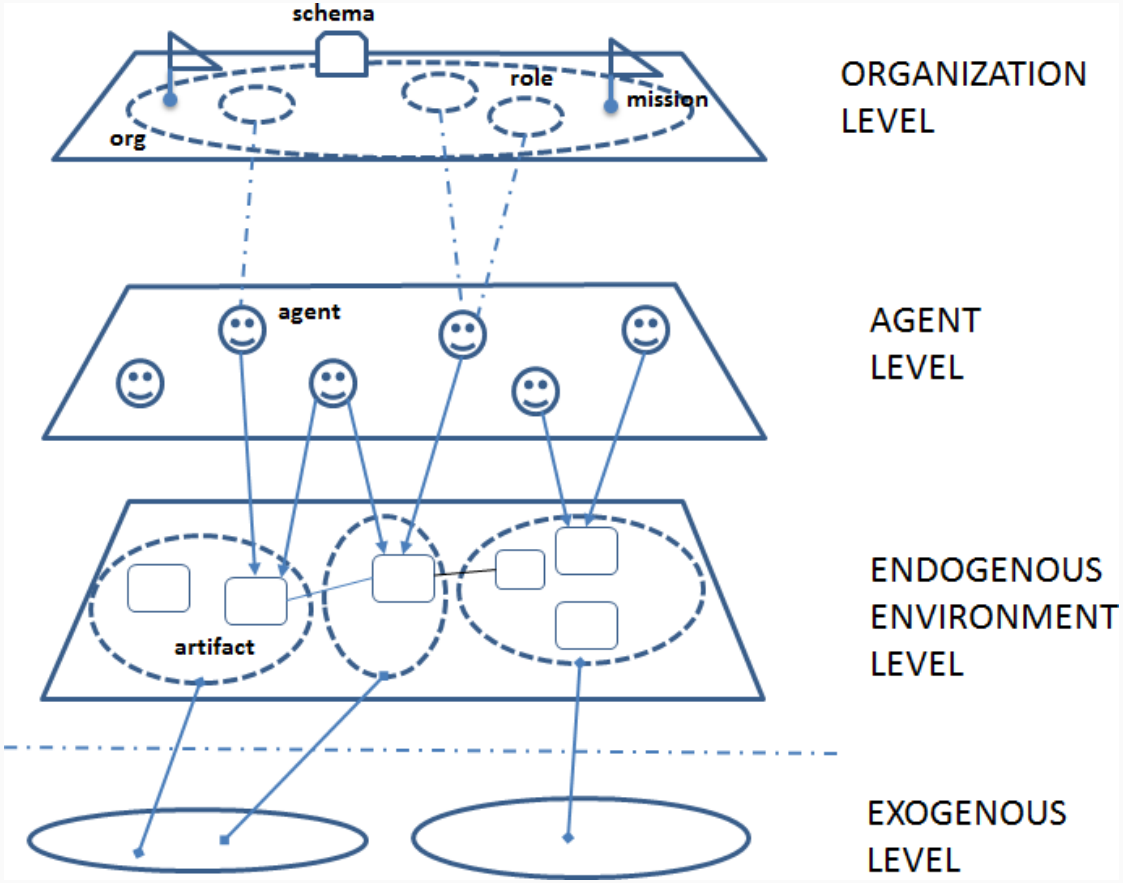
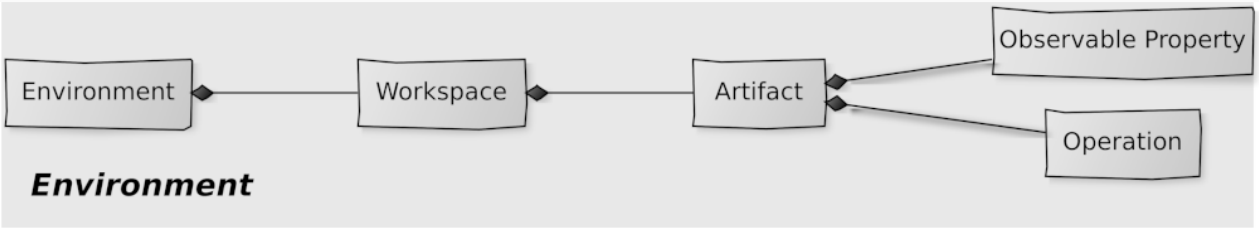


Figure 2: MAOP levels [Boissier et al., 2019]

Environment dimension



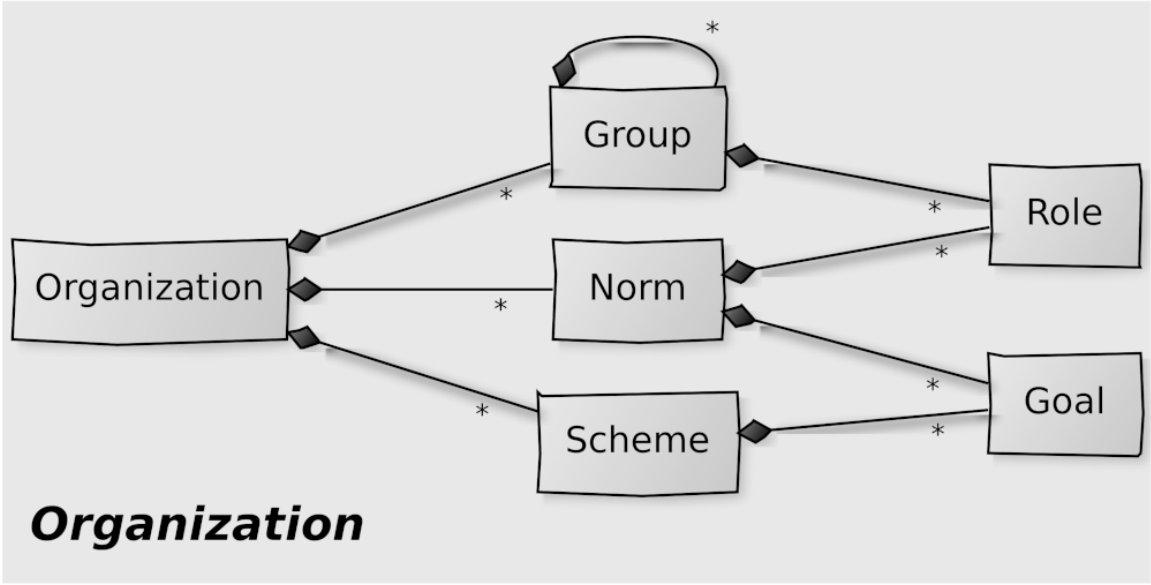
Simplified conceptual view (A&A meta-model [Omicini et al., 2008])

Simple artifact program:

```
public class Counter extends Artifact {
    void init(int initialValue) {
        defineObsProperty("count", initialValue);
    }

    @OPERATION void inc() {
        ObsProperty prop = getObsProperty("count");
        prop.updateValue(prop.intValue()+1);
    }
}
```

From [Boissier et al., 2019]

Organization dimension



Simplified Conceptual View ($\mathcal{M}$oise meta-model [Hübner et al., 2009])

Excerpts from organisation program:

```
<structural-specification>

<role-definitions>
 <role id="auctioneer" />
 <role id="participant" />
</role-definitions>

<group-specification id="auctionGroup">
 <roles>
  <role id="auctioneer"  min="1" max="1"/>
  <role id="participant" min="0" max="300"/>
 </roles>
</group-specification>
</structural-specification>
```

Structural spec.

```
<functional-specification>
 <scheme id="doAuction">
   <goal id="auction">
     <argument id="Id" />
     <argument id="Service" />
     <plan operator="sequence">
       <goal id="start" />
       <goal id="bid"    ttf="10 seconds" />
       <goal id="decide" ttf="1 hour" />
     </plan>
   </goal>
   <mission id="mAuctioneer" min="1" max="1">
     <goal id="start" />
     <goal id="decide" />
   </mission>
```

Functional spec.

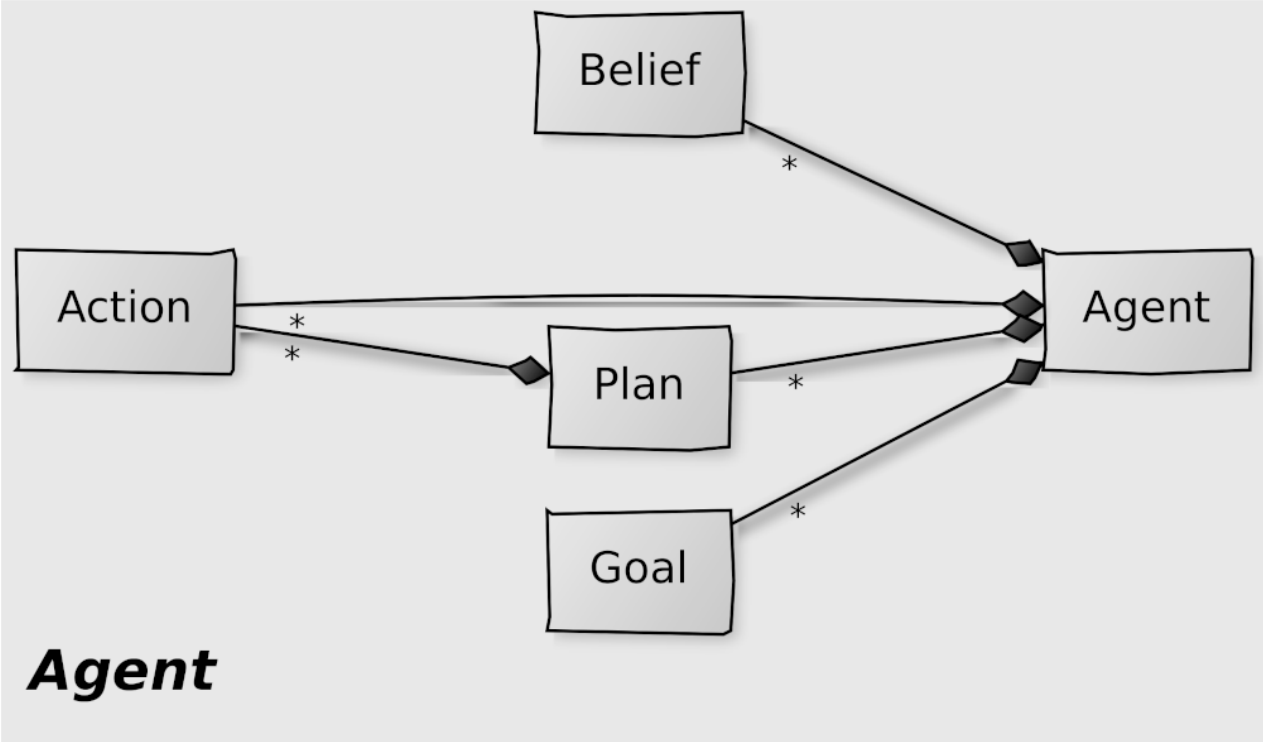
```
<normative-specification>
   <norm id="n1"  type="permission"
         role="auctioneer"
         mission="mAuctioneer" />
   <norm id="n2"  type="obligation"
         role="participant"
         mission="mParticipant" />
</normative-specification>
```

Normative spec.

```
norm n1 : plays(A, auctionneer, G) ->
  forbidden(A,n1,plays(A,participant,G),
    'forever').
```

program in NPL

Agent dimension



Simplified Conceptual View (Jason meta-model [Bordini et al., 2007]):

Simple Agent Program:

```
happy(bob). // initial belief
!say(hello). // initial goal
/* Plans */
+!say(X) : happy(bob) <- .print(X).
// ...
```

example bob.asl

```
+happy(A) <- !say(hello(A)).
+!say(A) : not today(friday) <- .print(X); !say(X).
+!say(X) : today(friday) <- .print("stop").
-happy(A) : .my_name(A) <- .drop_intention(say(_)).
```

example carl.asl

Agent's plans for obligations

Obligation to commit to a mission

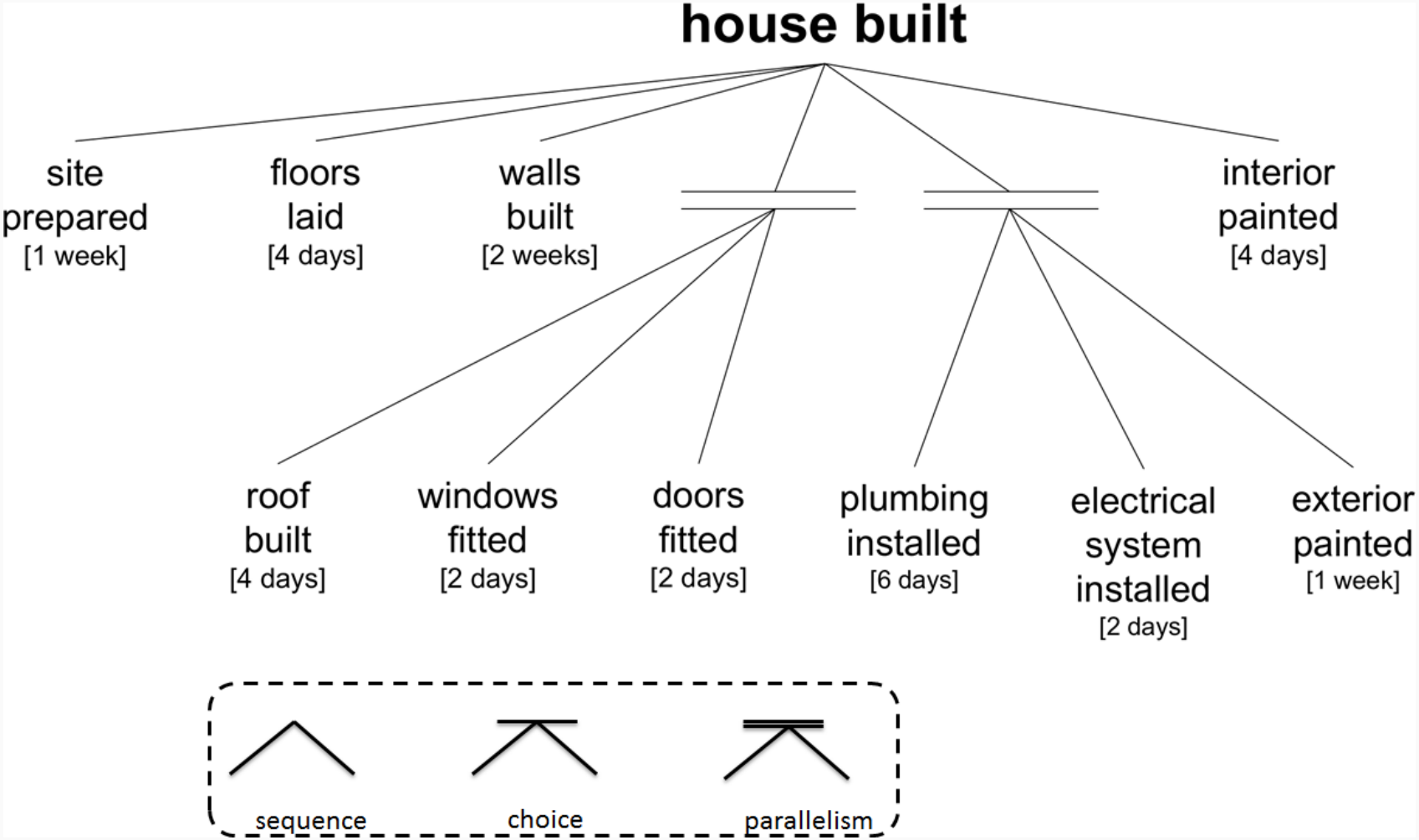
```
+obligation(Ag,Norm,committed(Ag,Mission,Scheme),DeadLine)
   : .my_name(Ag)
   <- .print("I am obliged to commit to ",Mission);
      commit_mission(Mission,Scheme).
```

Obligation to a goal

```
+obligation(Ag,Norm,achieved(Sch,Goal,Ag),DeadLine)
   : .my_name(Ag)
   <- .print("I am obliged to achieve goal ",Goal);
      !Goal[scheme(Sch)];
      goal_achieved(Goal,Sch).
```

The builing house example

Functional decomposition



Extending JaCaMo for Accountability

- Changes mostly concern the organization specification; i.e., the Moise component
- Changes are as conservative as possible (when no accountability relationship is specified, we fall back to standard JaCaMo)
- Changes satisfy three needs:
  - specify accountability agreements within an organization
  - translate accountability agreements into a corresponding body of norms
  - give agents the capability of producing accounts and marking goals not only as *achieved*, but also as *failed* or *released*

Extending JaCaMo for Accountability

Specifying Accountability Agreement

- The functional spec. is extended to include an XML tag for accountability agreement

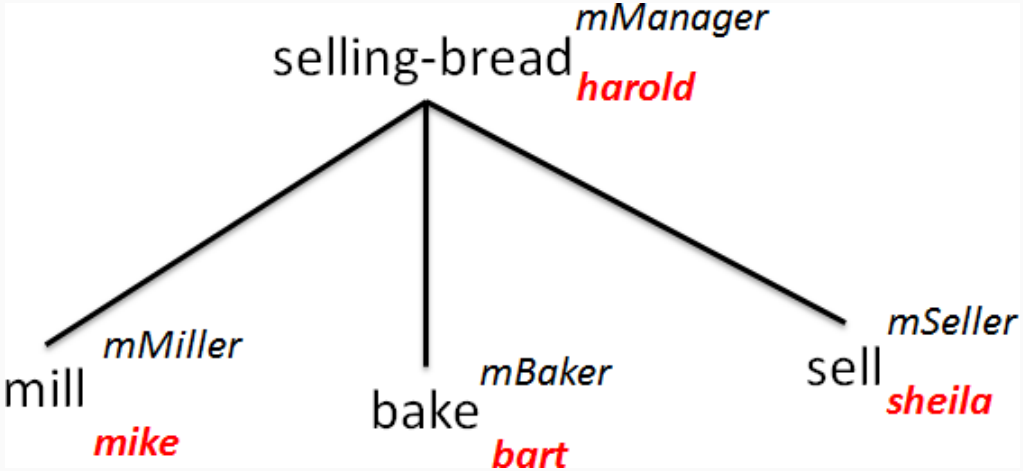
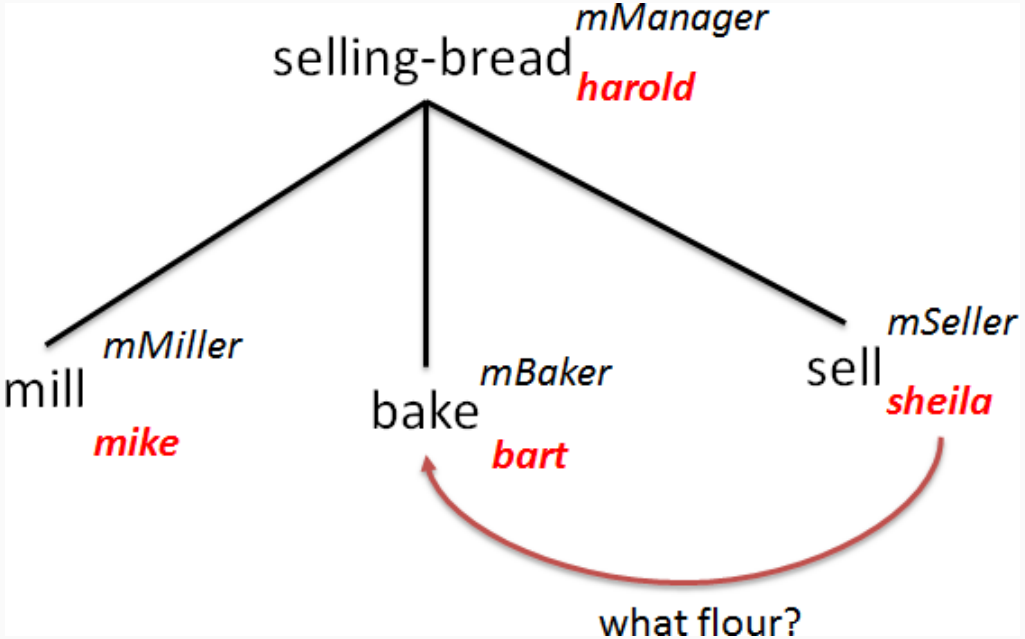
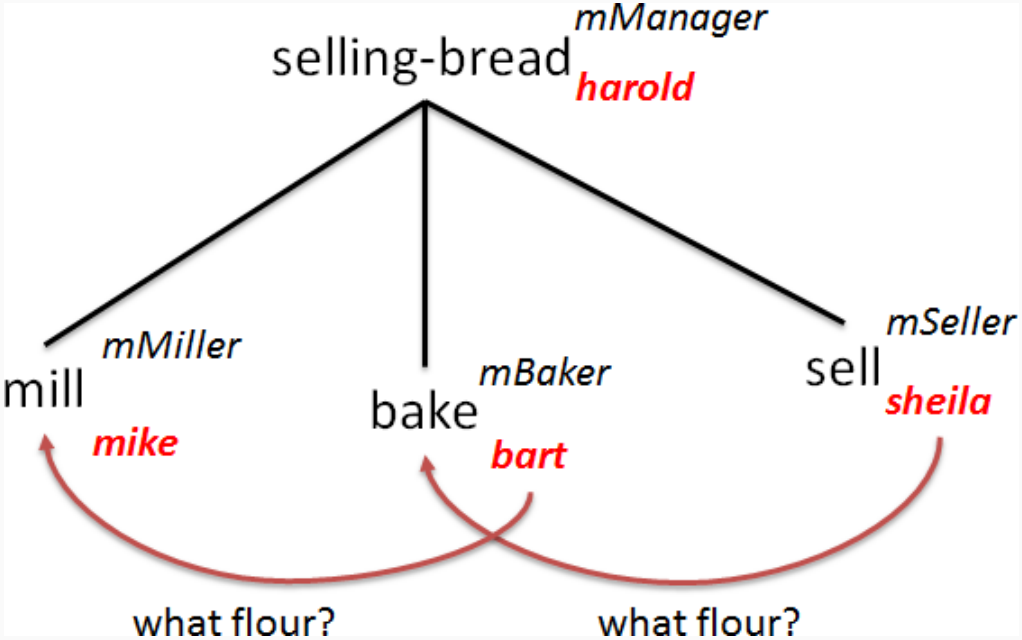
```
Accountability Agreement id: aa1
  concerns: u
  can request when: r

  Account Template:
    requesting goal: TASK
    accountability goal: TASK
       must account with: arguments
    treatment goal: TASK
       must treat when: condition
```

```
<supervision-strategy id="strParcel">
    <accountability-agreement id="aaParcel">
        <target id="reachDestination" />
        <request-template id="reqStock">
            <requesting-condition value="delay" />
            <goal id="requestDelayReason" />
        </request-template>
        <account-template id="accParcel">
            <goal id="reportDelayReason" />
            <account-argument id="reason" arity="1" />
            <account-argument id="closedRoads" arity="1" />
        </account-template>
    </accountability-agreement>
    <treatment-policy id="tpParcel">
        <treatment-condition value="true" />
        <goal id="updateMap" />
    </treatment-policy>
</supervision-strategy>
```

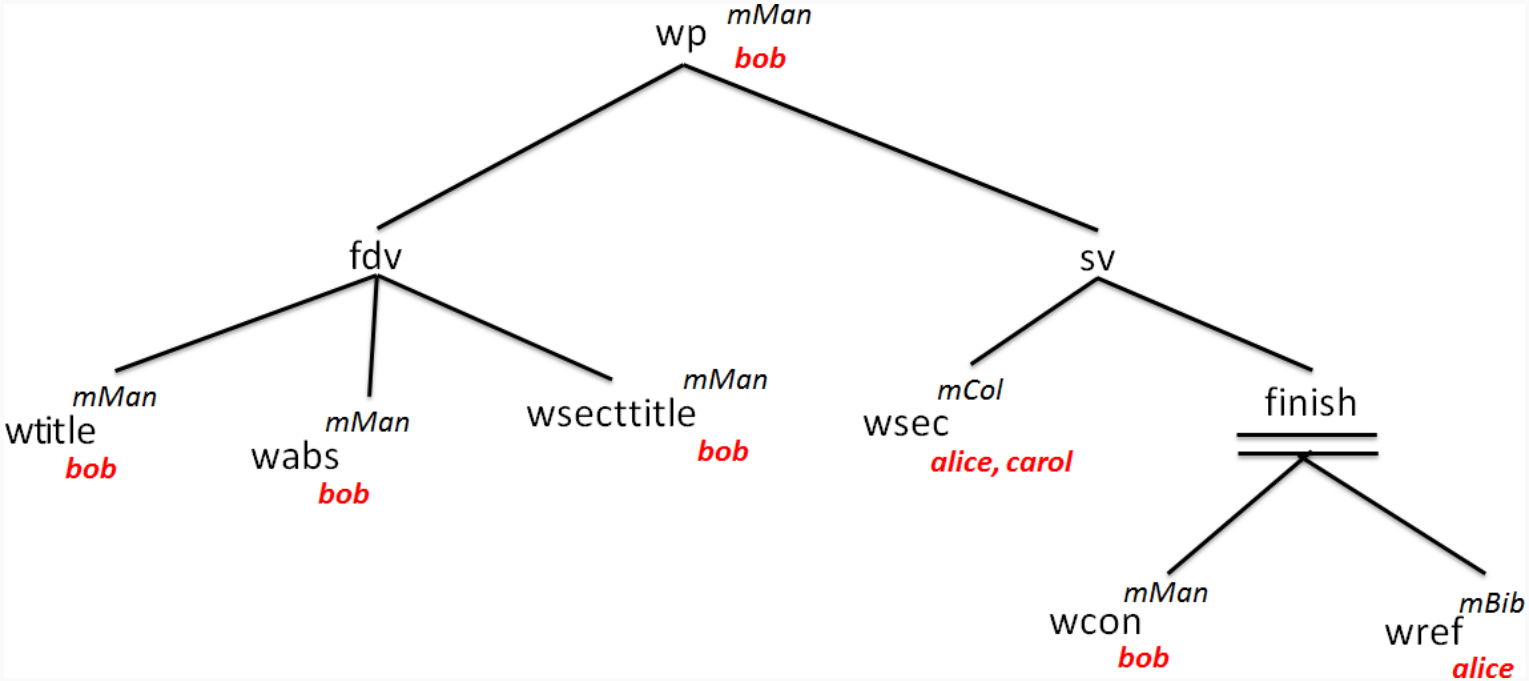
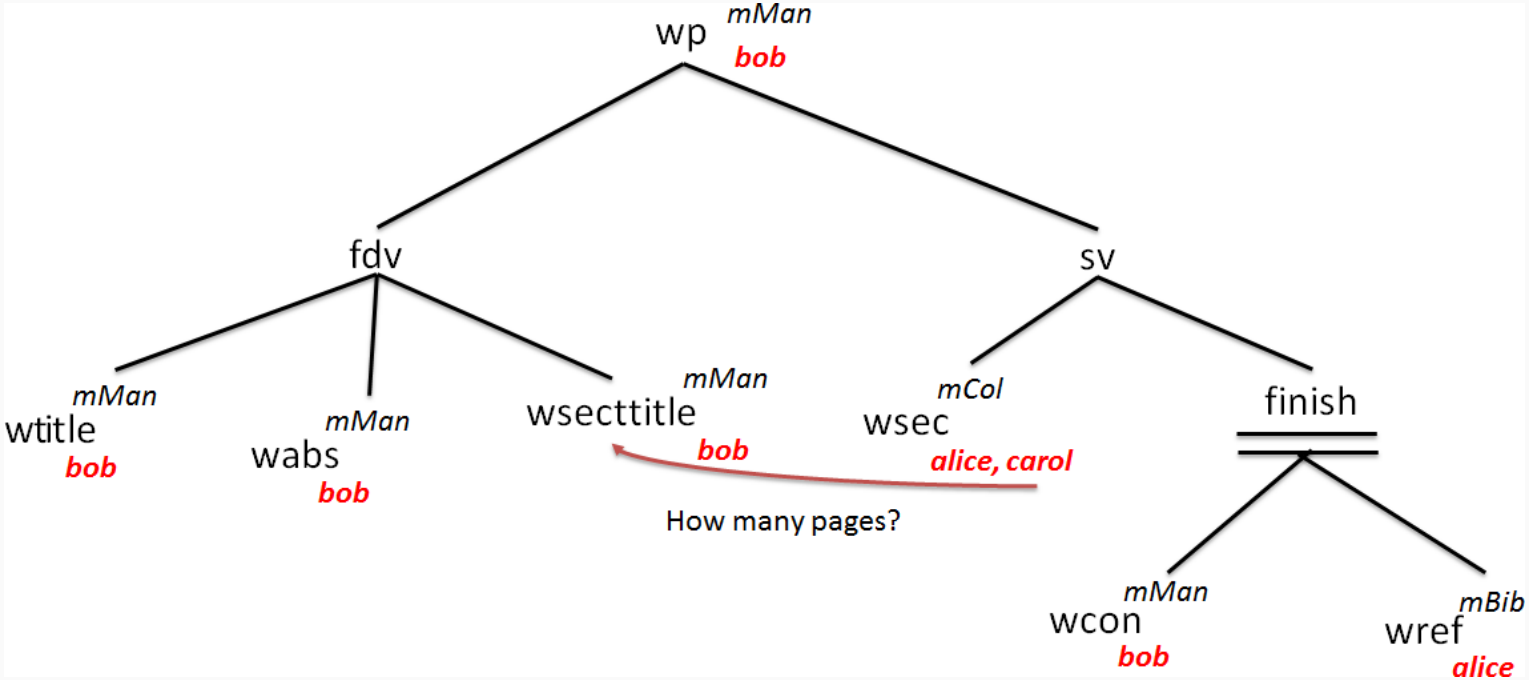
Extending JaCaMo for Accountability

Generating norms from agreements

- New normative facts, such as:
  - `requestingGoal(G)`
  - `accountabilityAgreement(id, target)`
  - `requestTemplate(id, condition, requesting goal)`
  - `accountTemplate(id, accounting goal)`
  - `accountArgument(account template id, functor, arity)`
  - ...
- New norms and rules, such as:
  - ```
    enabled(S,G) :-
    goal(_, G,  dep(or,PCG), _, NP, _) & requestingGoal(G) &
    requestTemplate(RT,Condition,G) & Condition & NP \== 0 &
    any_satisfied(S,PCG).
    ```
    ```

Extending JaCaMo for Accountability

Generating accounts

- This extension touches the artifact representing organizational schema
- New operations are made available to the agents:
  - `giveAccount()`: generates an account as an observable property within the artifact (publicly accessible)
  - `goalFailed()`: marks a goal as failed: the organizational goal cannot proceed
  - `goalReleased()`: marks a goal as released: even though the goal has not been satisfied (it may be even failed), the goal is no longer required (usually after an appropriate treatment), and hence the organizational goal can be resumed

Where to find the software...

- JaCaMo home page (standard version by Boissier et al.)
  http://jacamo.sourceforge.net/
- JaCaMo for accountability
  http://di.unito.it/moiseaccountability
- JaCaMo for exception handling
  https://sourceforge.net/projects/moise-exceptions/

Note: No need to install standard JaCaMo first, our projects are self-contained

Accountability in Practice

How does accountability come into play while programming agents?

Three basic programming patterns can be exploited for capturing a variety of situations

- Information gathering
- Context-aware adaptation
- Exception handling

Information gathering

Objective: let an agent gather information at runtime to complete its goals

```
+!organizational-goal
    : <account missing>
    <-
      goalAchieved(requestForAccount);
      .wait(500);
      !organizational-goal.

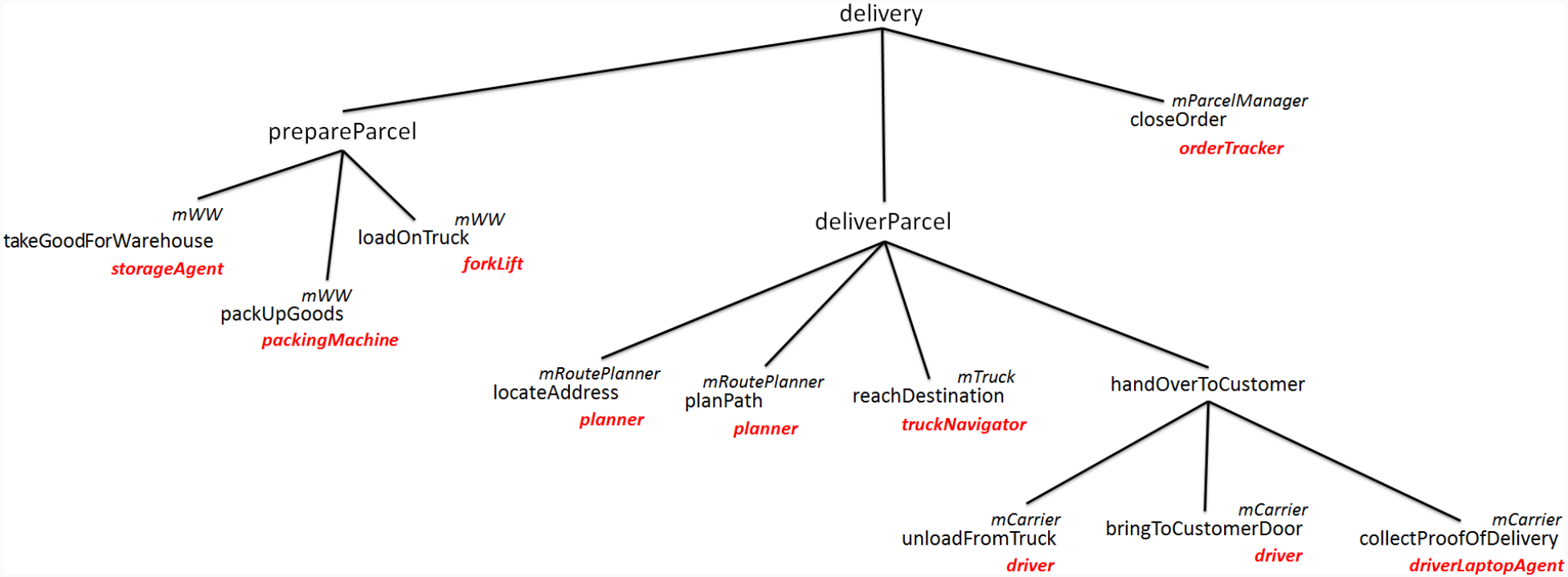
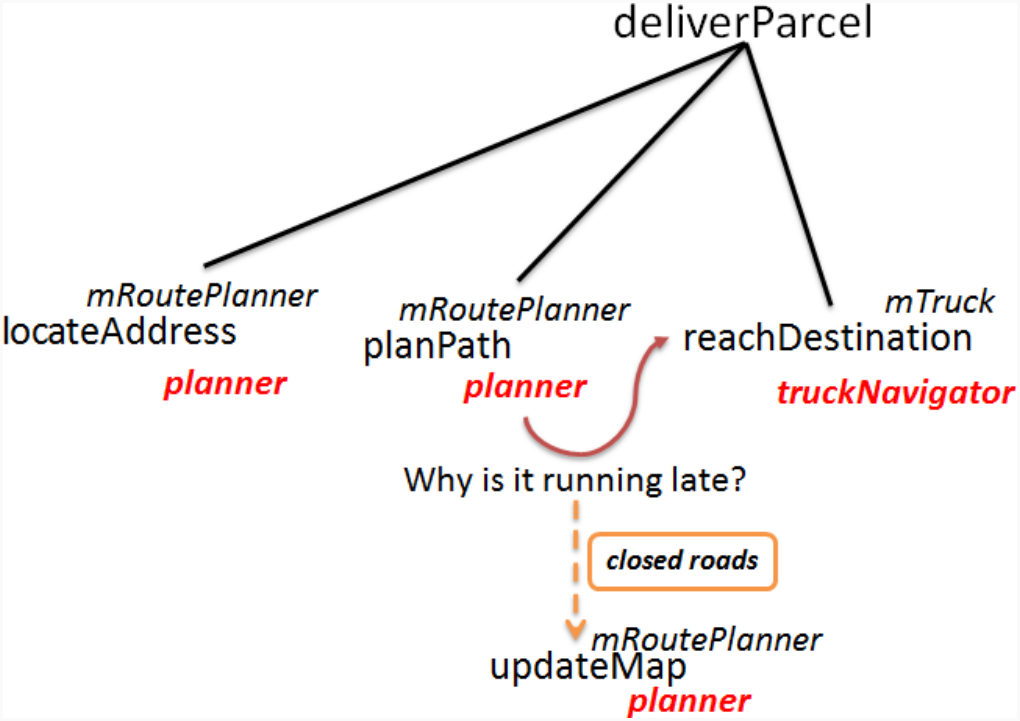
+!organizational-goal
    : <account ready>
    <-
      <do something useful>
      goalAchieved(organizational-goal).
```

Information gathering for decision making

Scenario: Harold is the owner of an organization produces and sells bread



- Sheila can decide, autonomously, the price of the bread she sells.
- She can fix a reasonable price on her own, or she can exploit infos about the production process; e.g., is the flour "standard" or organic?

Information gathering for decision making

Relying on an accountability relationship, Sheila could ask an account to Bart about the bake goal.



Information gathering for decision making

In turn, Bart could ask an account to Mike about goal `mill`.



Information gathering for decision making

Conceptually, the accountability agreement between Sheila and Bart is:

```
Accountability Agreement id: aa1
  concerns: bake
  can request when: true

  Account Template:
    requesting goal: requestFlourType
    accountability goal: notifyFlourType
       must account with: flourType
```

The *a-taker* and *a-giver* agents are designated by including goal `requestFlourType` in a Sheila's mission, and goal `notifyFlourType` in a Bart's mission, respectively

Information gathering for decision making

Looking at the XML specifying the organization...

```
<supervision-strategy id="strWhatFlour">
      <accountability-agreement id="aa1">
        <target id="bake" />
        <request-template id="reqWhatFlour">
            <requesting-condition value="true" />
            <goal id="requestFlourType"  type="achievement"/>
        </request-template>
        <account-template id="accFT">
           <goal id="notifyFlourType" />
           <account-argument id="flourType" arity="1" />
        </account-template>
     </accountability-agreement>
   </supervision-strategy>
   ...
    <mission id="mBaker" min="1" max="1">
        <goal id="bake"/>
        <goal id="notifyFlourType"/>
    </mission>
    <mission id="mSeller" min="1" max="1">
        <goal id="sell"/>
        <goal id="requestFlourType"/>
    </mission>
    ...
```

Between Bart and Mike a similar agreement exists

Information gathering for decision making

Applying the pattern to Sheila's program

```
+!sell :
 not account(flourType(N))
    <-
  .print("I need more info to fix the price!");
  goalAchieved(requestFlourType);
  .print("asked for the type of flour");
  .wait(500);
  !sell.


+!sell :
   account(flourType(N))
<-
  if (N == "bio"){
   .println("Bio flour set high price at 9.00 euros");
 }
 else {
   .println("Standard flour set low price at 2.00 euros");
 }.
```

DEMO LIVE

Information gathering for integration

Scenario: Bob, Alice and Carol cooperate for writing a paper (see JaCaMo documentation)



Alice and Carol are both asked to achieve goal `wsec`. However, they achieve the goal in different ways:

- Carol needs to know how many pages must be written
- Alice writes as many pages as she can

Information gathering for integration

Let us assume the following agreement exists



Information gathering for integration

Conceptually, the following accountability agreement is specified between Carol and Bob

```
Accountability Agreement id: aa2
  concerns: wsecttitle
  can request when: true

  Account Template:
    requesting goal: requestSectionLength
    accountability goal: notifySectionLenth
       must account with: pageNumber
```

Information gathering for integration

Applying the pattern to Carol's program

```
+!wsecs[scheme(S)]
: not account(pageNumber(N))
<-
  .print("asking for page number");
  goalAchieved(requestSectionLength);
  .wait(500);
  !wsecs[scheme(S)].


+!wsecs[scheme(S)]
: account(pageNumber(N))
 <-
 .print("writing ", N, " pages for scheme 2 ",S,"...").
```

DEMO LIVE

Context-aware adaptation

Objective: Acquire contextual conditions that are of some interest and that require a special treatment when they occur.

The treatment is part of the organization specification

$\Rightarrow$ *A proper obligation will be issued*

```
+newInterestingSituation
: <some-context>
<-
  goalAchieved(requestForAccount).

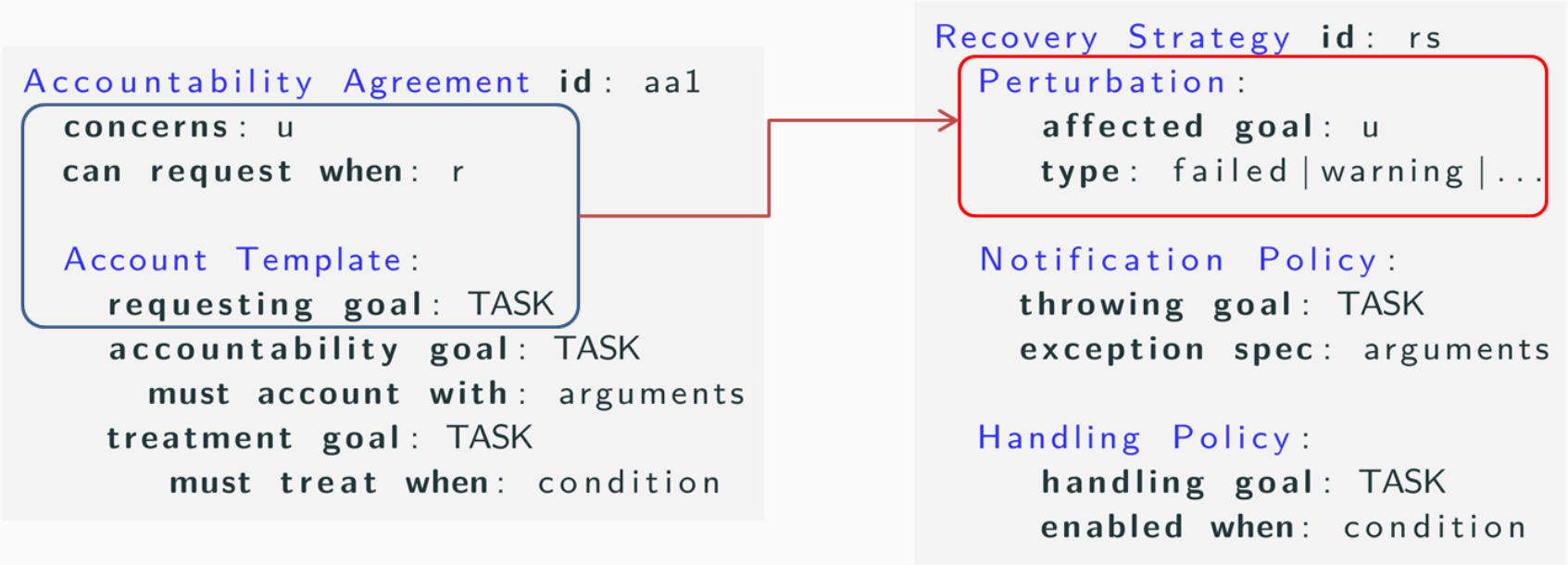
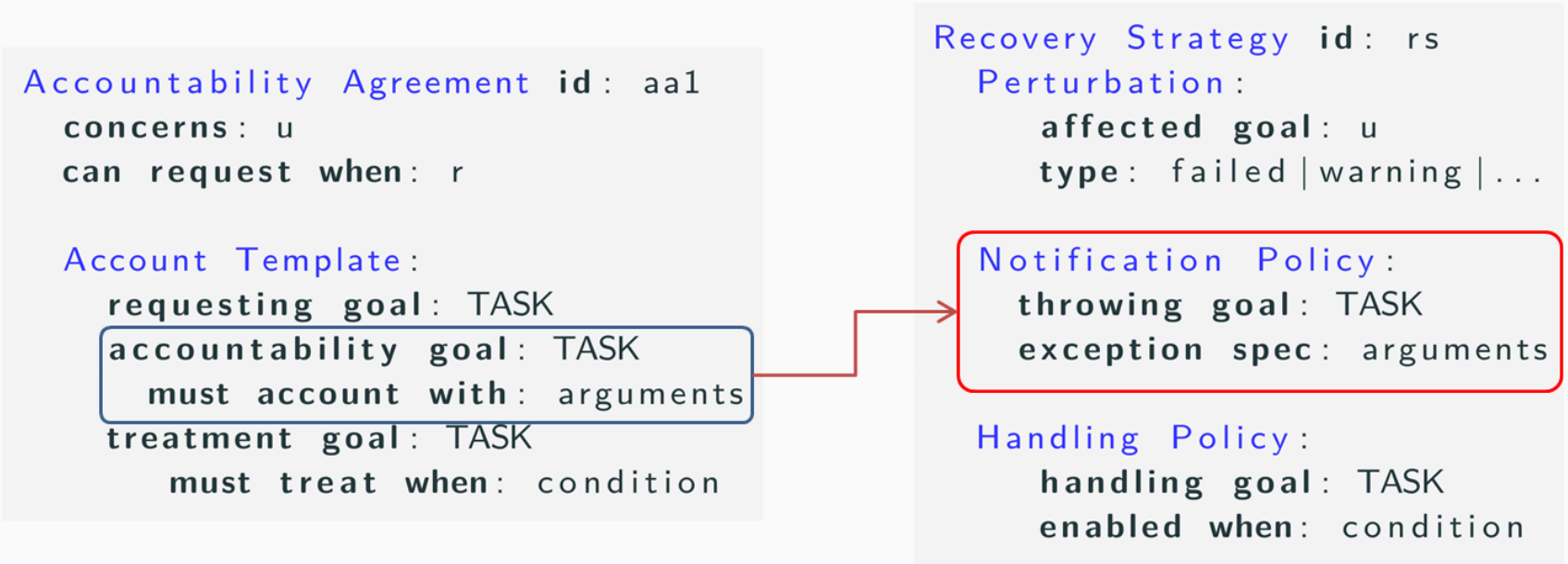
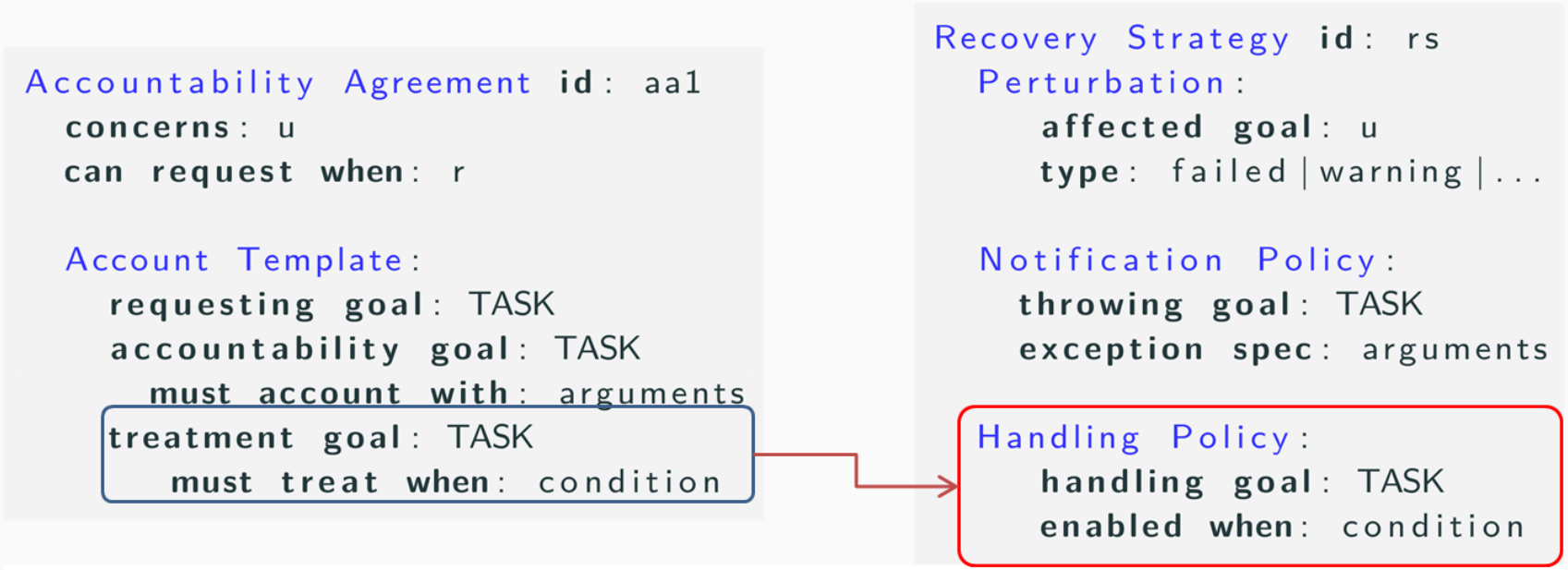
+!treatNewInterestingSituation
: <account-about-the-situation>
<-
  <do-something-about-the-situation>;
  goalAchieved(treatNewInterestingSituation).
```

Context-aware adaptation

Scenario: The delivery of some goods involves many agents from the packaging to the shipping. *Many things can go wrong!*



Context-aware adaptation

- Reaching a destination could require longer than expected due to some closed road.
- Updating the planner's map could prevent delays in the next shipping



Information gathering for integration

Conceptually, the accountability agreement is structured as follows

```
Accountability Agreement id: aaParcel
  concerns: reachDestination
  can request when: true

  Account Template:
    requesting goal: requestDelayReason
    accountability goal: reportDelayReason
       must account with: reason
       must account with: closedRoads
    treatment goal: updateMap
       must treat when: true
```

Context-aware adaptation

Applying the pattern to the `planner` agent

```
+oblUnfulfilled(obligation(_,_,done(_,reachDestination,_),_))
  <- println("*** REQUESTING DELAY REASON ***");
     goalAchieved(requestDelayReason).

+!updateMap
     : account(closedRoads(I))
  <- println("*** ADDING CLOSED ROADS TO IGNORE LIST... ***");
     +ignore(I).
```

- **Accountability promotes adaptation/innovation**:
  The planner now excludes the closed roads while planning the routes of future shipping
- **Accountability is not blame**:
  Sanctioning the truck would not allow the planner to know why the delivery is delayed, future deliveries would be affected by the same problem

DEMO LIVE

Exceptions and Exception Handling

Exception

Event that causes suspension of normal program execution. [ISO/IEC/IEEE, 2010]

- The purpose of an exception handling mechanism is to:
  1. Identify when an **exception** (i.e., a *perturbation*) occurs
  2. Apply suitable **handlers**, capable of treating the exception and recover

Responsibility in Exception Handling

Exception handling is a matter of **responsibility distribution**:

1. Always involves two parties: a party that is **responsible** for raising an exception, and another party that is **responsible** for handling it
2. Captures the need for some **information/account** from the former to the latter that allows coping with the exception

Exception handling can be built upon accountability relationships

Accountability and Exception handling

- For the exception handling purpose, the account is naturally given automatically as soon as a perturbation occurs
- The request for an account is implicit (the a-taker may not even observe the perturbation so it could not ask for an account)

A change of perspective

Terminologically

- a-giver raises an exception: exception raiser
- a-taker handles an exception: exception handler
- an account is mapped into the raised exception

Conceptually

```
Accountability Agreement id: aa1
  concerns: u
  can request when: r

  Account Template:
    requesting goal: TASK
    accountability goal: TASK
      must account with: arguments
    treatment goal: TASK
       must treat when: condition
```

```
Recovery Strategy id: rs
  Perturbation:
     affected goal: u
     type: failed|warning|...

  Notification Policy:
    throwing goal: TASK
    exception spec: arguments

  Handling Policy:
     handling goal: TASK
     enabled when: condition
```

A change of perspective

Terminologically

- a-giver raises an exception: exception raiser
- a-taker handles an exception: exception handler
- an account is mapped into the raised exception

Conceptually



The occurrence of a perturbation is associated with an implicit request for account

A change of perspective

Terminologically

- a-giver raises an exception: exception raiser
- a-taker handles an exception: exception handler
- an account is mapped into the raised exception

Conceptually



Providing an account amount to raising an exception

A change of perspective

Terminologically

- a-giver raises an exception: exception raiser
- a-taker handles an exception: exception handler
- an account is mapped into the raised exception

Conceptually



The treatment of an account corresponds to handling the raised exception

A change of perspective

At the normative level

- An *Accountability Agreement* yields
  - a **Permission** to request an account
  - an **Obligation** to provide an account upon request
  - in same cases, there may also be an **Obligation** to treat the account
- A *Recovery Strategy* yields
  - an **Obligation** to raise an exception upon the detection of a perturbation
  - an **Obligation** to handle the raised exception

Our JaCaMo extension needs further refinements to capture such a semantics

Extending JaCaMo for Exception Handling

Generating norms from agreements

- New normative facts, such as:
  - `failureReason(F)`
  - `throwException(G,[errorCode(F)])[artifact_id(ArtId)])`
  - `exceptionThrown(S,G,)`
  - `exceptionArgument(S,G,errorCode(F))`
  - ...
- New norms and rules, such as:
  - ```
    enabled(S,TG) :- policy_goal(P,TG) &
       notificationPolicy(P,Condition) & Condition &
        goal(_, TG,  Dep, _, NP, _) & NP \== 0 &
        ((Dep = dep(or,PCG)  & (any_satisfied(S,PCG) |
              all_released(S,PCG))) |
        (Dep = dep(and,PCG) & all_satisfied_released(S,PCG))).
    ```
    ```

A pattern for Exception Handling

Raising-agent's side

```
//reacting to obligation permits to mark
//an organizational goal as failed
+obligation(Ag,_,done(_,SOME_GOAL,Ag),_)
  : my_name(Ag)
   <- !SOME_GOAL;
      goalAchieved(SOME_GOAL).

// organizational goal internalized by the agent
+!SOME_GOAL
   <- INTERNAL_GOAL.

//in case the internal goal fails, also the organizatior
//goal fails
-!SOME_GOAL[env_failure_reason(F)]
    <-
      +failureReason(F)
      goalFailed(site_prepared)[artifact_id(ArtId)];
      .fail.

//the agent is then asked to raise an exception
+obligation(Ag,_,done(_,RAISE_EX_SOME_GOAL,Ag),_)
    : .my_name(Ag) &
      failureReason(F)
   <-
      throwException(SOME_GOAL,[errorCode(F)])[artifact_id(ArtId)];
      goalAchieved(RAISE_EX_SOME_GOAL).
```

```
Recovery Strategy id: rs1
  Perturbation:
    affected goal: SOME_GOAL
    type: failed

  Notification Policy:
    throwing goal: RAISE_EX_SOME_GOAL
    exception spec: F

  Handling Policy:
    handling goal: HANDLING_GOAL
    enabled when: condition
```

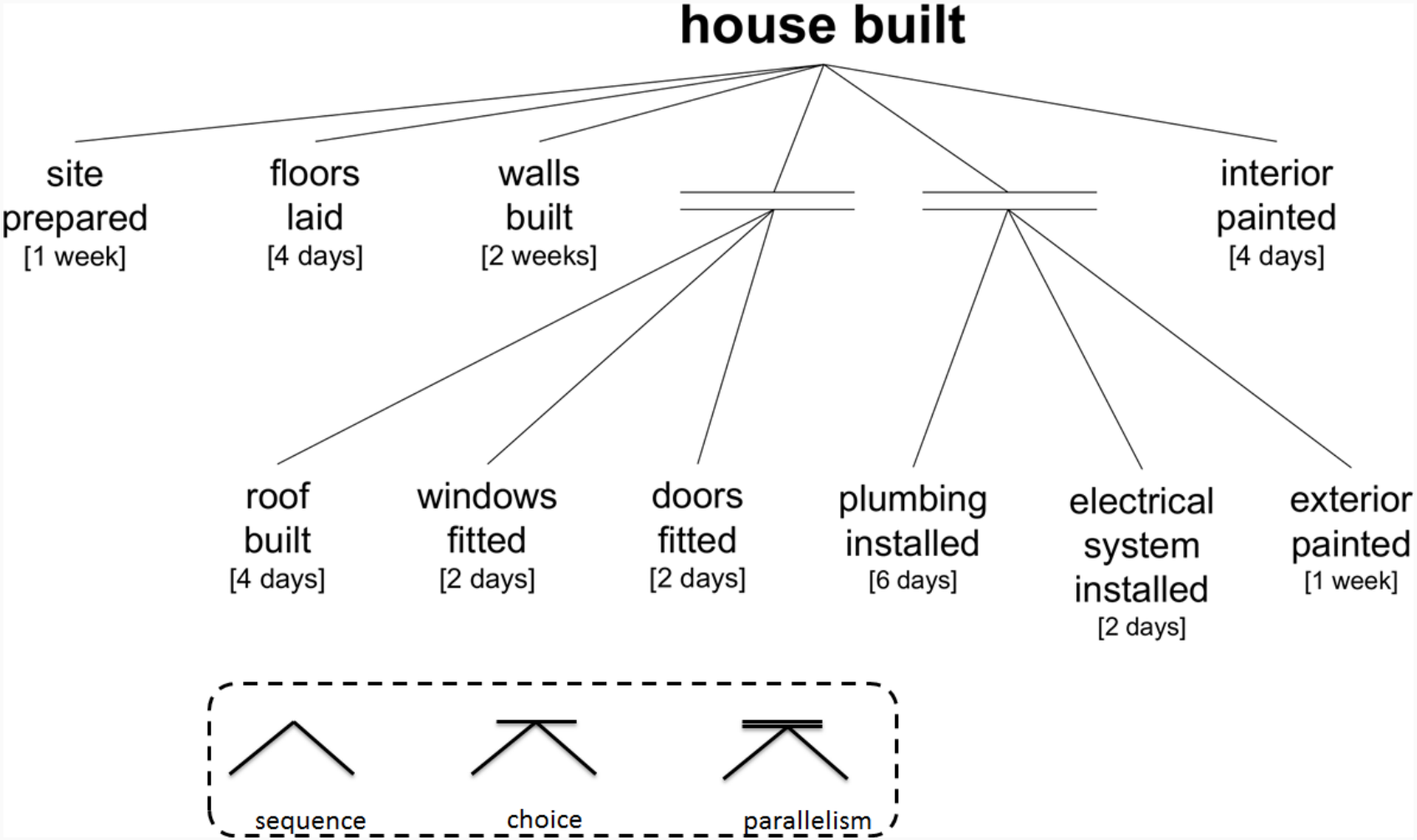
A pattern for Exception Handling

Handling-agent's side

```
//The handling agent react to the obligation of handlin~
//the exception relying on the information received
//along with the exception itself
+obligation(Ag,_,done(_,HANDLING_GOAL,Ag),_)
    : .my_name(Ag) &
      exceptionThrown(S,SOME_GOAL,_) &
      exceptionArgument(S,SOME_GOAL,errorCode(F))
   <-
      <do something useful to handle the exception>


      //The programmer decides how to consider the fail
      //Two alternatives:
      //[
          //The failed goal is released: no longer necessary
          goalReleased(SOME_GOAL)[artifact_id(ArtId)];

          //OR

          //the failed goal is reset: try once more to get it done
          resetGoal(SOME_GOAL)[artifact_id(ArtId)]
      //]

      //the handling goal is set achieved
      goalAchieved(HANDLING_GOAL).
```

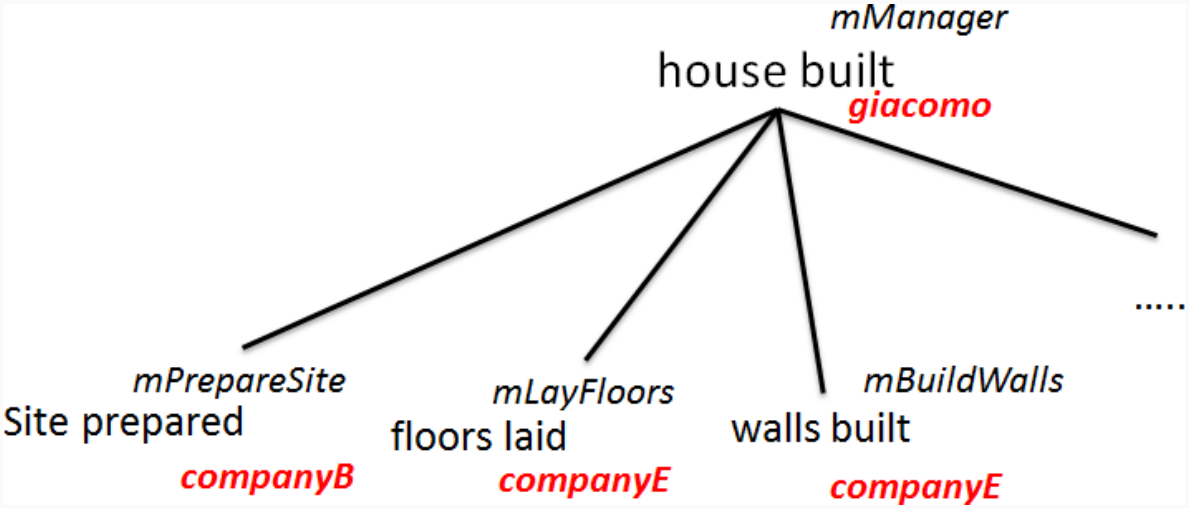
```
Recovery Strategy id: rs1
  Perturbation:
    affected goal: SOME_GOAL
    type: failed

  Notification Policy:
    throwing goal: RAISE_EX_SOME_GOAL
    exception spec: F

  Handling Policy:
    handling goal: HANDLING_GOAL
    enabled when: condition
```

Adding exception handling to Building House

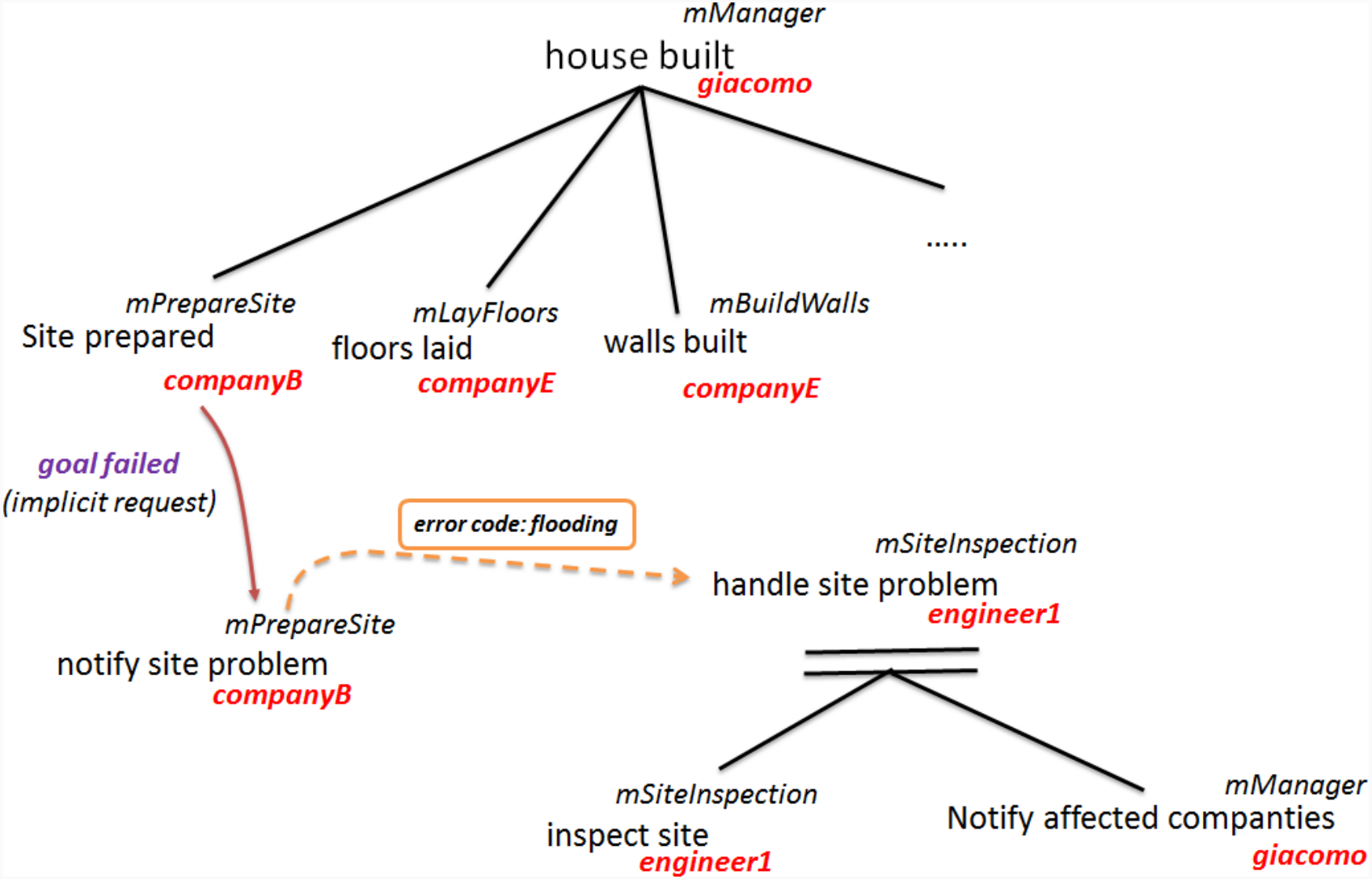


Adding exception handling to Building House

Scenario: Giacomo sets up an organization for building his house. Many companies joins the organization contributing on specific goals.



Adding exception handling to Building House



Adding exception handling to Building House

The following recovery strategy captures such a scenario

```
Recovery Strategy id: rsSitePreparation
  Perturbation:
     affected goal: site_prepared
     type: failed

  Notification Policy:
    throwing goal: site_preparation_exception
    exception spec: errorCode

  Handling Policy:
     handling goal: handle_site_problem
     enabled when: true
```

Adding exception handling to Building House

Applying the pattern to `companyB` (raising an exception on `site_prepared`)

```
+obligation(Ag,_,done(_,site_prepared,Ag),_)
  : my_name(Ag)
   <- !site_prepared;
      goalAchieved(site_prepared).

+!site_prepared
   <- println("Preparing site...");
      .wait(2000);
      prepareSite. // simulates the action (in GUI artifact)

-!site_prepared[env_failure_reason(F)]
    : focused(ora4mas,bhsch,ArtId)
   <- println("The site is flooded due to ",F,"!");
      +failureReason(F)
      goalFailed(site_prepared)[artifact_id(ArtId)];
      .fail.

+obligation(Ag,_,done(_,notify_site_preparation_problem,Ag),_)
    : .my_name(Ag) &
      focused(ora4mas,bhsch,ArtId) &
      failureReason(F)
   <- println("THROWING SITE PREPARATION EXCEPTION WITH ERROR CODE ",F,"!")
      throwException(site_preparation_exception,[errorCode(F)])[artifact_id(ArtId)];
      -failureReason(F);
      goalAchieved(notify_site_preparation_problem).
```

Adding exception handling to Building House

Applying the pattern to `engineer1` (handling the exception)

```
+obligation(Ag,_,done(_,inspect_site,Ag),_)
    : .my_name(Ag) &
      focused(ora4mas,bhsch,ArtId) &
      exceptionThrown(bhsch,site_preparation_exception,_) &
      exceptionArgument(bhsch,site_preparation_exception,errorCode(flooding))
   <- println("Inspecting site...");
      .wait(2000);
      performSiteAnalysis(Result);
      println("Done!");
      println("Fixing flooding...");
      .wait(2000);
      fixFlooding(Result);
      println("Done!");
      goalReleased(site_prepared)[artifact_id(ArtId)];
      goalAchieved(inspect_site).

+obligation(Ag,_,done(_,inspect_site,Ag),_)
    : .my_name(Ag) &
      focused(ora4mas,bhsch,ArtId) &
      exceptionThrown(bhsch,site_preparation_exception,_) &
      exceptionArgument(bhsch,site_preparation_exception,errorCode(archaeologicalRemains)
   <- println("Inspecting site...");
      .wait(2000);
      delimitSite;
      println("Done!");
      println("RemovingRemains...");
      .wait(2000);
      carefullyRemoveRemains;
      println("Done!");
      resetGoal(site_prepared)[artifact_id(ArtId)].
```

Modularity via Exception Handling

Exception Handling as a means for Modularity

Exceptions and exception handling mechanisms are not needed to deal just with errors.

They are needed, in general, as a means of conveniently interleaving actions belonging to different levels of abstraction. [Goodenough, 1975]

- Exceptions allow the invoker of an operation to extend the operation domain (the set of inputs for which effects are defined), or its range (the effects obtained when certain inputs are processed)
- Increase in the generality of an operation: the appropriate "fixup" will depend on the invoker's objectives
- **By grounding exception handling on accountability, we meet this vision!**

Outline of a General Methodology

1. Specify a (standard) JaCaMo organization:
   roles, groups, functional decomposition, missions
2. Extend the organization by means ***accountability relationships***:
   - **Accountability Agreements**
     - What roles may need information for its decision making process
     - What other roles may supply the needed information
     - Individuate requesting and notifying goals
     - Assign these goals to the missions of a-takers and a-givers
   - **Exception Handling** (recovery strategy)
     - What goals can fail and for what reasons (exception)
     - What countermeasures are available (exception handling goals)
     - How a perturbation should be notified (exception raising goals)
     - Assign these goals to the missions of handlers and raisers
3. Apply the patterns for programming the agents

Note: In principle both accountability agreements and recovery strategies could be defined within the same organization

CONCLUSIONS

Accountability for Software Engineering

Adopting **Accountability/Exception Handling as a design principle** induces several advantages:

- allows a designer to take into account the problem of **robustness since the early stages**, by providing a clear framework for specifying perturbations and the components dedicated to their handling
- promotes the **modularity and reuse** of software
- promotes **incrementality and integration** of new software
- maintains a **clear separation** between what it is expected the system do (i.e., functional decomposition), and how the system adapts itself to perturbations or contextual changes
- promotes **innovation** by way of the gained flexibility

Acknowledgments

- We are very grateful to **Olivier Boissier**, **Rafael H. Bordini**, **Federico Capuzzimati**, **Jomi F. Hübner**, **Katherine M. May**, and **Alessandro Ricci** for the very stimulating and insightful discussions about JaCaMo and Accountability
- A special thank to **Stefano Tedeschi** for having implemented the JaCaMo extension
- The AThOS[1] project for funding the initial steps of this adventure

[1] Accountable Trustworthy Organizations and Systems

References

Auditor General of Canada (2002).
Report of the Auditor General of Canada to the House of Commons, Chapter 9 Modernizing Accountability in the Public Sector.
Technical report, Office of the Auditor General of Canada.

Alderson, D. L. and Doyle, J. C. (2010).
Contrasting views of complexity and their implications for network-centric infrastructures.
IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 40(4).

Anderson, P. A. (1981).
Justifications and precedents as constraints in foreign policy decision- making.
American Journal of Political Science, 25(4).

Baldoni, M., Baroglio, C., Boissier, O., May, K. M., Micalizio, R., and Tedeschi, S. (2018a).

Accountability and Responsibility in Agents Organizations.
In Miller, T., Oren, N., Sakurai, Y., Noda, I., Savarimuthu, T., and Son, T. C., editors, *PRIMA 2018: Principles and Practice of Multi-Agent Systems, 21st International Conference*, number 11224 in Lecture Notes in Computer Science, pages 403–419, Tokyo, Japan. Springer.


Baldoni, M., Baroglio, C., Boissier, O., Micalizio, R., and Tedeschi, S. (2020a).
Accountability and Responsibility in Multiagent Organizations for Engineering Business Processes.
In Dennis, L. A., Bordini, R. H., and Lespèrance, Y., editors, *Post-Proc. of the 7th International Workshop on Engineering Multi-Agent Systems, EMAS 2019, Revised Selected Papers*, number 12058 in LNAI, pages 3–24, Montreal, QC, Canada. Springer.


Baldoni, M., Baroglio, C., Boissier, O., Micalizio, R., and Tedeschi, S. (2021a).

Distributing Responsibilities for Exception Handling in JaCaMo.
In Emdriss, U., Nowé, A., Dignum, F., and Lomuscio, A., editors, *Proc. of 20th International Conference on Autonomous Agents and Multiagent Systems, Demonstration Track, AAMAS 2021*, Online. IFAAMAS, Richland, SC.


Baldoni, M., Baroglio, C., Boissier, O., Micalizio, R., and Tedeschi, S. (2021b).
Exception Handling in Multiagent Organizations: Playing with JaCaMo.
In Alechina, N., Baldoni, M., and Logan, B., editors, *Proc. of the 9th International Workshop on Engineering Multi-Agent Systems, EMAS 2021, held in conjuction with AAMAS 2021*, Online.


Baldoni, M., Baroglio, C., Capuzzimati, F., and Micalizio, R. (2018b).
Commitment-based Agent Interaction in JaCaMo+.
Fundamenta Informaticae, 159(1-2):1–33.


Baldoni, M., Baroglio, C., Capuzzimati, F., and Micalizio, R. (2018c).
Type Checking for Protocol Role Enactments via Commitments.
Journal of Autonomous Agents and Multi-Agent Systems, 32(3):349–386.


Baldoni, M., Baroglio, C., May, K. M., Micalizio, R., and Tedeschi, S. (2016).
Computational Accountability.
In Chesani, F., Mello, P., and Milano, M., editors, *Deep Understanding and Reasoning: A challenge for Next-generation Intelligent Agents, URANIA 2016*, volume 1802, pages 56–62, Genoa, Italy. CEUR, Workshop Proceedings.


Baldoni, M., Baroglio, C., May, K. M., Micalizio, R., and Tedeschi, S. (2018d).
Computational Accountability in MAS Organizations with ADOPT.

Applied Sciences, 8(4).


Baldoni, M., Baroglio, C., May, K. M., Micalizio, R., and Tedeschi, S. (2019a).
MOCA: An ORM MOdel for Computational Accountability.
Journal of Intelligenza Artificiale, 13(1):5–20.


Baldoni, M., Baroglio, C., and Micalizio, R. (2018e).
The AThOS Project: First Steps towards Computational Accountability.
In Baldoni, M., Baroglio, C., and Micalizio, R., editors, *Proc. of the First Workshop on Computational Accountability and Responsibility in Multiagent Systems, CARe-MAS 2017*, volume 2051, Nice, France. CEUR, Workshop Proceedings.


Baldoni, M., Baroglio, C., and Micalizio, R. (2019b).
Accountability, Responsibility and Robustness in Agent Organizations.
In Dignum, V., Noriega, P., and Verhagen, H., editors, *Proc. of the 1st International Workshop on Responsible Artificial Intelligence*

Agents, RAIA 2019, held in conjuction with AAMAS 2019, Montreal, Canada.


Baldoni, M., Baroglio, C., and Micalizio, R. (2020b). **Fragility and Robustness in Multiagent Systems.** In Baroglio, C., Hubner, J. F., and Winikoff, M., editors, *Post-Proc. of the 8th International Workshop on Engineering Multi-Agent Systems, EMAS 2020, Revised Selected Papers*, number 12589 in LNAI, pages 61–77, Auckland, New Zealand. Springer.


Baldoni, M., Baroglio, C., Micalizio, R., and Tedeschi, S. (2020c). **Is Explanation the Real Key Factor for Innovation?** In Musto, C., Magazzeni, D., Ruggieri, S., and Semeraro, G., editors, *Proc. of Italian Workshop on Explainable Artificial Intelligence, XAI.it 2020*, volume 2742, pages 87–95, Online Event, Italy. CEUR, Workshop Proceedings.


Baldoni, M., Baroglio, C., Micalizio, R., and Tedeschi, S. (2021c). **Robustness based on Accountability in Multiagent Organizations.**

In Emdriss, U., Nowé, A., Dignum, F., and Lomuscio, A., editors, *Proc. of 20th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2021*, pages 142–150, Online. IFAAMAS, Richland, SC.


Baldoni, M., Baroglio, C., Micalizio, R., and Tedeschi, S. (2021d). **Social Commitments for Engineering Interaction in Distributed Systems.**
In Kalech, M., Abreu, R., and Last, M., editors, *Artificial Intelligence Methods for Software Engineering*, chapter 3, pages 51–85. World Scientific Publishing Company.


Boissier, O., Bordini, R., J.F., H., and Ricci, A. (2019). **Multi-Agent Oriented Programming - A short and practical introduction to the JaCaMo platform.**


Boissier, O., Bordini, R. H., Hübner, J. F., Ricci, A., and Santi, A. (2013).
Multi-agent oriented programming with JaCaMo.
Science of Computer Programming, 78(6):747–761.

Bordini, R. H., Hübner, J. F., and Wooldridge, M. (2007).
***Programming multi-agent systems in AgentSpeak using Jason*.**
John Wiley & Sons.

Burgemeestre, B. and Hulstijn, J. (2015).
***Handbook of Ethics, Values, and Technological Design*, chapter Design for the Values of Accountability and Transparency, pages 303–333.**
Springer.

Button, G. and Sharrock, W. (1998).
The organizational accountability of technological work.
Social Studies of Science, 28(1):73–102.

Chopra, A. K. and Singh, M. P. (2014).
The thing itself speaks: Accountability as a foundation for requirements in sociotechnical systems.
In *IEEE 7th Int. Workshop RELAW*. IEEE Computer Society.

Darwall, S. (2013).

***Morality, Authority, and Law: Essays in Second- Personal Ethics I*, chapter Civil Recourse as Mutual Accountability.**
Oxford University Press.


Darwall, S. L. (2006).
***The Second-person Standpoint: Morality, Respect, and Accountability*.**
Harvard University Press, Cambridge, MA.


Dignum, V., Baldoni, M., Baroglio, C., Caon, M., Chatila, R., Dennis, L. A., Génova, G., Haim, G., Kliess, M., Lopez-Sanchez, M., Micalizio, R., Pavón, J., Slovkovik, M., Smakman, M., van Steenberg, M., Tedeschi, S., van der Torre, L., Villata, S., and de Wildt, T. (2018).
Ethics by Design: Necessity or Curse?
In Furman, J., Marchant, G., Price, H., and Rossi, F., editors, *Proc. of 2018 AAAI/ACM conference on Artificial Intelligence, Ethics, and Society (AIES 2018)*, pages 60–66, New Orleans, USA. ACM.


Dignum, V., Dignum, F., and Meyer, J.-J. (2004).

An agent-mediated approach to the support of knowledge sharing in organizations.
The Knowledge Engineering Review, 19(2):147–174.


Dubnick, M. J. (2013).
Blameworthiness, trustworthiness, and the second-personal standpoint: Foundations for an ethical theory of accountability.

Presented at EGPA Annual Conference, Group VII: Quality and Integrity of Governance, Edinburgh, Scotland.


Dubnick, M. J. and Justice, J. B. (2004).
Accounting for accountability.
Annual Meeting of the American Political Science Association.


Durkheim, E. (1893).
***De la division du travail social*.**


Elder-Vass, D. (2011).
***The Causal Power of Social Structures: Emergence, Structure and Agency*.**

Cambridge University Press.


European Commission (2018).
General Data Protection Regulation.
https://eur-lex.europa.eu/legal-content/IT/TXT/HTML/?uri=CELEX:32016R0679.


Feltus, C. (2014).
Aligning Access Rights to Governance Needs with the Responsability MetaModel (ReMMo) in the Frame of Enterprise Architecture.
PhD thesis, University of Namur, Belgium.


Garfinkel, H. (1967).
Studies in ethnomethodology.
Prentice-Hall Inc., Englewood Cliffs, New Jersey.


Goldberg, J. C. P. and Zipursky, B. C. (2010).
Torts as wrongs.
Texas Law Review, 88.

Goodenough, J. B. (1975).
Exception handling design issues.
SIGPLAN Not., 10(7):41–45.

Grant, R. W. and Keohane, R. O. (2005).
Accountability and Abuses of Power in World Politics.
The American Political Science Review, 99(1).

Hübner, J. F., Boissier, O., Kitio, R., and Ricci, A. (2010).
Instrumenting multi-agent organisations with organisational artifacts and agents.
Auton. Agents Multi Agent Syst., 20(3):369–400.

Hubner, J. F., Sichman, J. S., and Boissier, O. (2007).
Developing organised multiagent systems using the MOISE+ model: Programming issues at the system and agent levels.
Int. J. Agent-Oriented Softw. Eng., 1(3/4):370–395.

ISO/IEC/IEEE (2010).

ISO/IEC/IEEE International Standard - Systems and software engineering – Vocabulary.
ISO/IEC/IEEE 24765:2010(E), pages 1–418.


López y López, F. and Luck, M. (2003).
Modelling norms for autonomous agents.
In *4th Mexican International Conference on Computer Science (ENC 2003), 8-12 September 2003, Apizaco, Mexico*, pages 238–245. IEEE Computer Society.


Marengo, E., Baldoni, M., Baroglio, C., Chopra, A., Patti, V., and Singh, M. (2011).
Commitments with regulations: reasoning about safety and control in REGULA.
In *The 10th International Conference on Autonomous Agents and Multiagent Systems*, volume 2 of *AAMAS '11*, pages 467–474. IFAAMAS.


Parsons, T. (1968).
***The Structure of Social Action*.**

Collier-Macmillan, London.


Ricci, A., Piunti, M., Viroli, M., and Omicini, A. (2009).
Environment programming in cartago.
In Bordini, R. H., Dastani, M., Dix, J., and Seghrouchni, A. E. F., editors, *Multi-Agent Programming, Languages, Tools and Applications*, pages 259–288. Springer.


Romzek, B. S. and Dubnick, M. J. (1987).
Accountability in the public sector: lessons from the challenger tragedy.
Public Administration Review, 47(3):227–238.


Schlenker, B. R., Britt, a. W., Pennington, J., Rodolfo, M., and Doherty, K. (1994).
The triangle model of responsibility.
Psychological Review, 101(4):632–652.


Simon, J. (2015).

***The Onlife Manifesto: Being human in a hyperconnected era*, chapter Distributed Epistemic Responsibility in a Hyperconnected Era, pages 145–159.**
Springer Open.


Suchman, L. (1997).
***Discourse, Tools, and Reasoning: Essays on Situated Cognition*, chapter Centers of Coordination: A Case and Some Themes.**
Springer.


Sustainable Energy for All Initiative.
Accountability framework.


Vincent, N. A. (2011).
***Moral Responsibility*, volume 27 of *Library of Ethics and Applied Philosophy*, chapter A Structured Taxonomy of Responsibility Concepts.**
Springer.


Wooldridge, M. J. (2002).

Introduction to multiagent systems.
Wiley.


Yazdanpanah, V. and Dastani, M. (2016).
Distant group responsibility in multi-agent systems.
In *PRIMA 2016: Princiles and Practice of Multi-Agent Systems - 19th International Conference, Phuket, Thailand, August 22-26, 2016, Proceedings*, pages 261–278.


Zahran, M. (2011).
Accountability Frameworks in the United Nations System.
https://www.unjiu.org/sites/www.unjiu.org/files/jiu_document_files/product
notes/JIU%20Products/JIU_REP_2011_5_English.pdf.

UN Report.