# A Focus on Robustness with ARFIN organizations

WOA 2019, Parma

Matteo Baldoni, Cristina Baroglio, Roberto Micalizio, Stefano Tedeschi

Università degli Studi di Torino - Dipartimento di Informatica, Torino, Italy

# Multiagent Organizations (MAO) in a few words

- **MAOs**: strategies for decomposing complex organizational goals into simpler **sub-tasks**, allocating them to **roles**.
- Current models target open systems by allocating and enforcing rights and duties to agents about the tasks to realize.
- Agents' activities are choreographed by issueing obligations.

# Multiagent Organizations (MAO) in a few words

- **MAOs**: strategies for decomposing complex organizational goals into simpler **sub-tasks**, allocating them to **roles**.
- Current models target open systems by allocating and enforcing rights and duties to agents about the tasks to realize.
- Agents' activities are choreographed by issueing obligations.

- **Agents**: by adopting roles agents execute the corresponding tasks in a distributed, coordinated, and regulated fashion.
- Each **agent**:
  - carries out part of the organizational goal,
  - depends on the **collaboration** of others to perform its task.

# Agents lose sight of the overall process

- They are focussed on the achievement of the assigned sub-goals
- Ignore the place of their goals in the big picture
- Who should give restitution to whom?
- Who is interested in my activities ("*stakeholders*")?

# Agents lose sight of the overall process

- They are focussed on the achievement of the assigned sub-goals
- Ignore the place of their goals in the big picture
- Who should give restitution to whom?
- Who is interested in my activities ("*stakeholders*")?

**Consequences**

1. Agents may have the capability of achieving the assigned goals but in ways that do not fit into the requirements of the specific stakeholder
2. When agents fail, the interested parties have no explicit mechanism for sorting out what occurred, for a redress

# Agents lose sight of the overall process

- They are focussed on the achievement of the assigned sub-goals
- Ignore the place of their goals in the big picture
- Who should give restitution to whom?
- Who is interested in my activities ("*stakeholders*")?

**Consequences**

1. Agents may have the capability of achieving the assigned goals but in ways that do not fit into the requirements of the specific stakeholder
2. When agents fail, the interested parties have no explicit mechanism for sorting out what occurred, for a redress

**Something is missing ...**

# Goals achieved but no proof

- Agents who enter the organization are under the regulation of norms, that stipulate their rights and duties
- However, there is no guarantee that they will provide the **accompanying proofs**, that are induced by their assigned responsibilities.

# Goals achieved but no proof

- Agents who enter the organization are under the regulation of norms, that stipulate their rights and duties
- However, there is no guarantee that they will provide the **accompanying proofs**, that are induced by their assigned responsibilities.

**Something is missing ...**

# Direction: MAO need "agent responsabilization"

Need of introducing some explicit representation of some **relationships** agents have with the others, their mutual "dependences", and, more broadly, of the dependence of the organization on its members for what concerns the *realization of the business process*.

# What about Commitments?

$$C(x, y, r, u)$$

**A social commitment is a promise (to bring about) from debtor to creditor**

The creditor should be the interested party but the choice to create the commitment and towards whom is totally up to the debtor.

# What about Commitments?

$$C(x, y, r, u)$$

**A social commitment is a promise (to bring about) from debtor to creditor**

The creditor should be the interested party but the choice to create the commitment and towards whom is totally up to the debtor.

**We need a different kind of relationship**

An agreement between the parts, respecting a specification inside an organization, whereby the **legitimacy** for one agent to ask information about a subgoal is accepted by **both** the involved agents

We resort on the notions of **responsibility** and **accountability**.

# Accountability

**(Dubnick and Justice, 2004)**

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

**(Dubnick and Justice, 2004)**

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 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**)."

# Accountability and Responsibility

As a consequence, accountability is **grounded on perceived/assumed responsibility**, deriving from **recognition of legitimacy** of exercising some power, and of the claim-right to hold the responsible to account.

# Accountability and Robustness

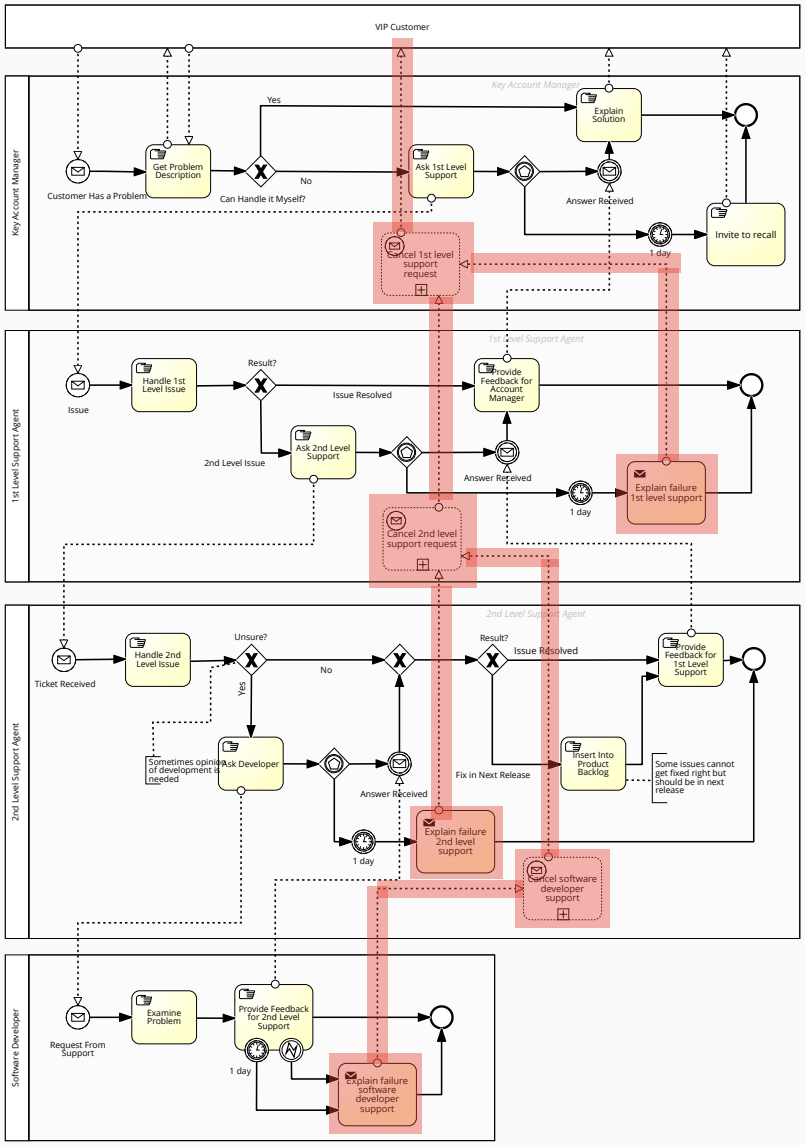
- **Accountable software**: software that, under given conditions, provides account of what was achieved or what went wrong.
- System results to be **robust**, that is capable **to keep on working** within **acceptable standards** despite something abnormal occurs.

# Exception handling

Exception handling as straightforward special case of accountability, where the agents have the agreement that the account-taker is **always** interested in **feedback**, on occurrence of some exceptions. Thereby, the account-giver proactively provides such feedback **without waiting for a request**

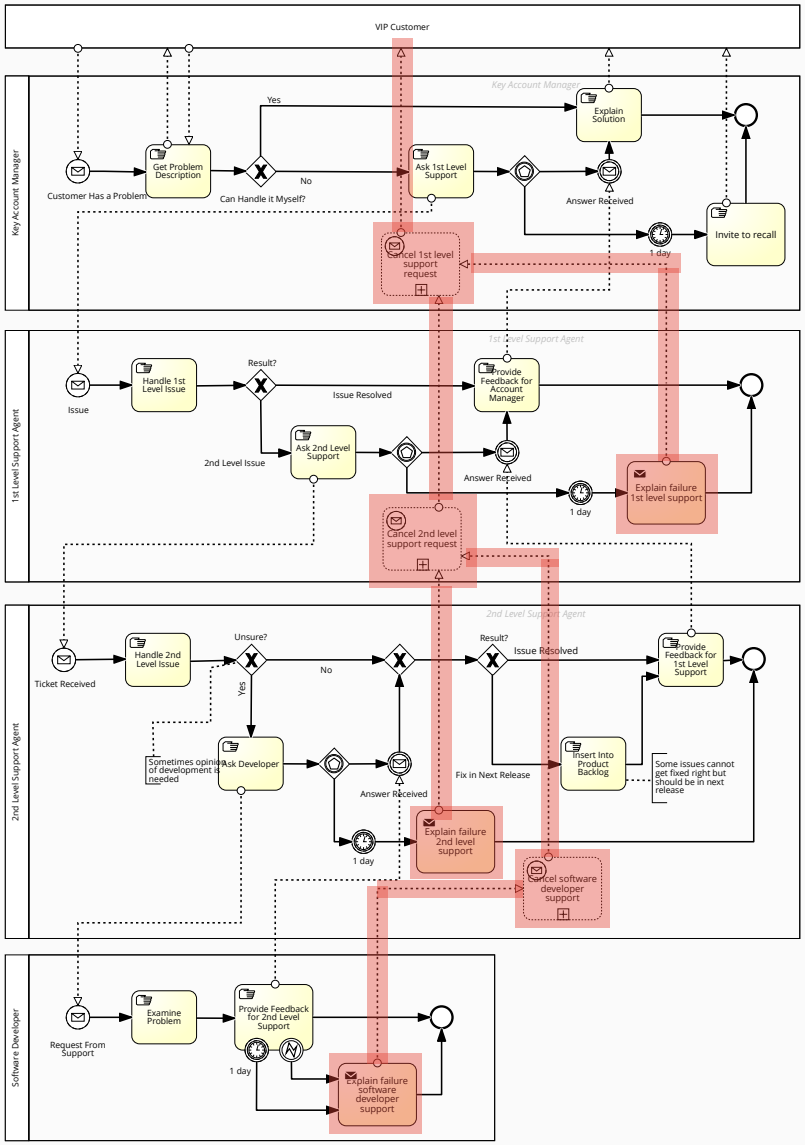
# Exception Handling: a special case of accountability

- Exception specification mechanism captures the way in which a process is interested into another



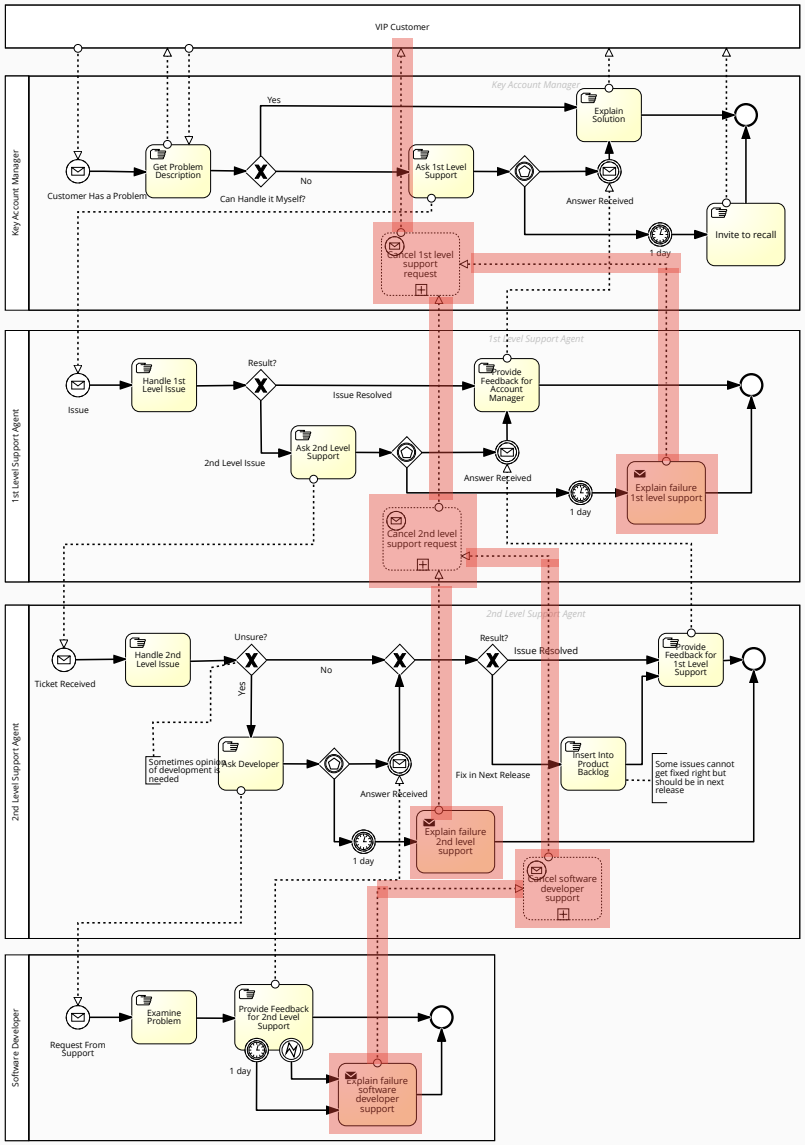
# Exception Handling: a special case of accountability

- Exception specification mechanism captures the way in which a process is interested into another
- A process can react to abnormal events (exceptions), possibly encompassing contextual information provided by others in their decision processes



# Exception Handling

- The driver of such a process is the attempt to execute up to the preset standards, possibly through self-regulation, by adapting either the execution or the organization itself
- This process heavily relies on the accounts that the involved agents are expected to produce.



# ARFIN Organizations

**Agent organization**

A process being collectively executed by a number of agents. Agents produce and answer to institutional events, and need to coordinate to accomplish the organizational goal.

**ARFIN organization**

A MAO that includes: an **accountability specification**, a **responsibility distribution**, an **accountability fitting**, and some **norms**.

# JaCaMo + A/R (Baldoni et al., 2018)

**JaCaMo + A/R**

(Baldoni et al., 2018) proposes to complement the specification of an organization with accountability and responsibility specifications.

# JaCaMo + A/R (Baldoni et al., 2018)

**JaCaMo + A/R**

(Baldoni et al., 2018) proposes to complement the specification of an organization with accountability and responsibility specifications.

Such an extension provides organizations with an additional infrastructure that captures who **should** give account to whom for certain states of the organization, and who **can** ask for such feedbacks.

# Specifying accountabilities

**Accountability** $\mathrm{A}(x, y, r, u)$

$x$, the account-giver, is accountable towards $y$, the account-taker, for the condition $u$ when the condition $r$ (*context*) holds.

**Accountability specification**

It is a set **A** of accountabilities $\mathrm{A}(x, y, r, u)$.

$\mathbb{A}$ denotes a set of accountability specifications.

# Specifying accountabilities

**Accountability** $\mathrm{A}(x, y, r, u)$

$x$, the account-giver, is accountable towards $y$, the account-taker, for the condition $u$ when the condition $r$ (*context*) holds.

**Accountability specification**

It is a set **A** of accountabilities $\mathrm{A}(x, y, r, u)$.

$\mathbb{A}$ denotes a set of accountability specifications.

*Accountability is grounded on control and expectation*:

- expectation is naturally conveyed with the accountability itself;
- (knowlegde) control is recursively verified on the structure of $u$: $x$ controls $u$ either directly (it is in position of causing $u$) or indirectly by relying on accountabilities.

# Specifying responsibilities

**Responsibility specification**

A responsibility specification $\mathrm{R}(x, q)$ expresses an expectation on any agent playing role $x$ on pursuing condition $q$ ($x$ is entitled and should have the capabilities of bringing about $q$).

**Responsibility assumption**: for playing role $x$ an agent should declare to accept to be considered in the position of causing $q$.

# Specifying responsibilities

**Responsibility specification**

A responsibility specification $\mathrm{R}(x, q)$ expresses an expectation on any agent playing role $x$ on pursuing condition $q$ ($x$ is entitled and should have the capabilities of bringing about $q$).

**Responsibility assumption**: for playing role $x$ an agent should declare to accept to be considered in the position of causing $q$.

**R** denotes a responsibility distribution, that is a set of responsibility assumptions.

# Accountability Fitting

**Accountability Fitting $\mathbf{R} \rightsquigarrow \mathbb{A}$ ("$\mathbf{R}$ fits $\mathbb{A}$" )**

Given:

- $\mathbb{A}$: a set of accountability specifications;
- $\mathbf{R}$: a responsibility distribution;

We say that $\mathbf{R} \rightsquigarrow \mathbb{A}$ when $\exists\ \mathbf{A} \in \mathbb{A}$ such that $\forall\ \mathrm{A}(x, y, r, u) \in \mathbf{A}$, $\exists\ \mathrm{R}(x, q) \in \mathbf{R}$ such that, for some actualization $\widehat{q}$, $(u/r)/\widehat{q} \equiv \top$.

# Accountability Fitting

**Accountability Fitting $\mathbf{R} \rightsquigarrow \mathbb{A}$ ("R fits $\mathbb{A}$" )**

Given:

- $\mathbb{A}$: a set of accountability specifications;
- **R**: a responsibility distribution;

We say that $\mathbf{R} \rightsquigarrow \mathbb{A}$ when $\exists\ \mathbf{A} \in \mathbb{A}$ such that $\forall\ \mathrm{A}(x, y, r, u) \in \mathbf{A}$, $\exists\ \mathrm{R}(x, q) \in \mathbf{R}$ such that, for some actualization $\widehat{q}$, $(u/r)/\widehat{q} \equiv \top$.

Given $\mathrm{R}(x, a \cdot b \cdot c)$, $\mathrm{A}(x, y, d \cdot e, d \cdot a \cdot c)$,
$q$ is $a \cdot b \cdot c$, $r$ is $d \cdot e$, $u$ is $d \cdot a \cdot c$,
then $(u/r)/\widehat{q}$ is $(d \cdot a \cdot c)/(d \cdot e)/(a \cdot b \cdot c)$.

# Accountability fitting

**Intuitively ...**

Accountability fitting captures a properly defined organization that is guaranteed to properly distribute responsibilities.

Not only the organization owns but it also to "connects" the needed, distributed control over the goal so as to better support its achievement.

# Accountability fitting

**Intuitively ...**

Accountability fitting captures a properly defined organization that is guaranteed to properly distribute responsibilities.

Not only the organization owns but it also to "connects" the needed, distributed control over the goal so as to better support its achievement.

An organization is properly specified when the *accountability fitting* $\mathbf{R} \rightsquigarrow \mathbb{A}$ holds.

# Accountability fitting

**Intuitively ...**

Accountability fitting captures a properly defined organization that is guaranteed to properly distribute responsibilities.

Not only the organization owns but it also to "connects" the needed, distributed control over the goal so as to better support its achievement.

An organization is properly specified when the *accountability fitting* $\mathbf{R} \rightsquigarrow \mathbb{A}$ holds.

$\mathbf{R} \rightsquigarrow \mathbb{A}$ provides a specification the agents **must explicitly conform to**, when enacting organizational roles.

# Accountability Fitting with exceptions

Given a set of events $\mathcal{U}$, let $\mathcal{E}$ be a set of exceptional events, that is, $\mathcal{E} \cap \mathcal{U} = \emptyset$ and each event $e \in \mathcal{E}$ is complementary to possibly many events in $\mathcal{U}$.

# Accountability Fitting with exceptions

Given a set of events $\mathcal{U}$, let $\mathcal{E}$ be a set of exceptional events, that is, $\mathcal{E} \cap \mathcal{U} = \emptyset$ and each event $e \in \mathcal{E}$ is complementary to possibly many events in $\mathcal{U}$.

$\mathcal{F} \subseteq \mathcal{U} \times \mathcal{E}$ maps events in $\mathcal{U}$ to their corresponding complementary ones in $\mathcal{E}$.

# Accountability Fitting with exceptions

Given a set of events $\mathcal{U}$, let $\mathcal{E}$ be a set of exceptional events, that is, $\mathcal{E} \cap \mathcal{U} = \emptyset$ and each event $e \in \mathcal{E}$ is complementary to possibly many events in $\mathcal{U}$.

$\mathcal{F} \subseteq \mathcal{U} \times \mathcal{E}$ maps events in $\mathcal{U}$ to their corresponding complementary ones in $\mathcal{E}$.

- An expression $u$ is *touched* by an exception $e \in \mathcal{E}$ if for at least one event $w$ occurring in $u$, $(w, e) \in \mathcal{F}$.
- An accountability relationships $\mathrm{A}(x, y, r, u)$ is *touched* by the occurrence of event $e$ when $w$ occurs in $u$ and $(w, e) \in \mathcal{F}$.

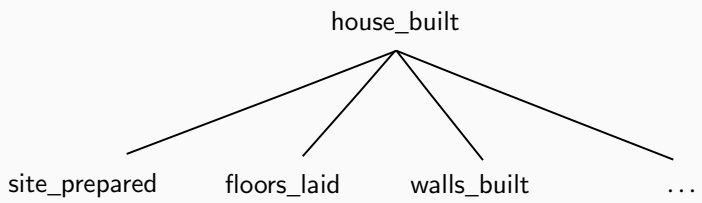
**Compliance with exceptions**

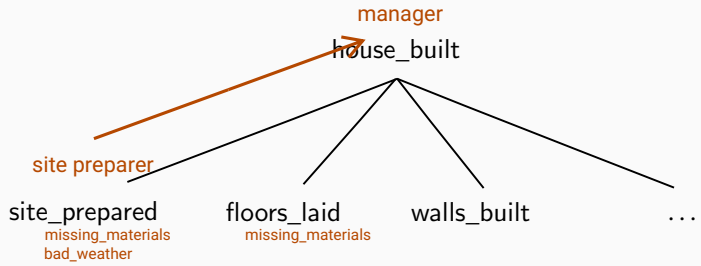
Let $[\mathbf{R} \rightsquigarrow \mathbf{A}]_{\mathcal{F}}$ be an accountability fitting characterized by $\mathcal{F}$. An ARFIN organization is compliant with $[\mathbf{R} \rightsquigarrow \mathbf{A}]_{\mathcal{F}}$ if, whenever $\mathrm{A}(x, y, r, u) \in \mathbf{A}$ is touched by an event $e \in \mathcal{E}$, an account about $u$ is requested to $x$ by default.

# MAO $\rightarrow$ ARFIN

Complementing a functional decomposition with an accountability fitting with exceptions $[\mathbf{R} \rightsquigarrow \mathbf{A}]_{\mathcal{F}}$ turns an organization (implemented in JaCaMo) into a particular kind of ARFIN organization that considers **abnormal situations explicitly**.

# The building house example



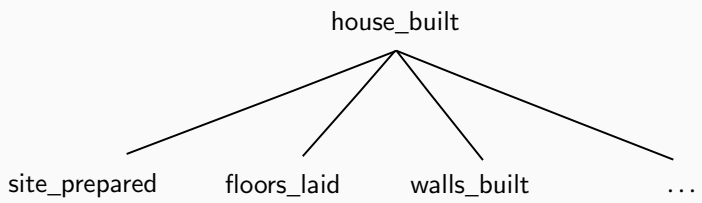



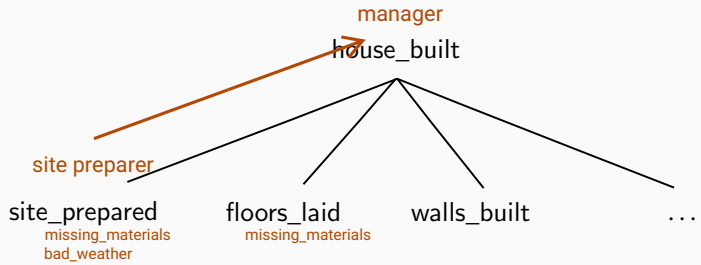

- R(*site preparer*, site_prepared) $\in$ **R**
- A(*site preparer*, *manager*, $\top$, site_prepared) $\in$ **A**

**Abnormal situations/exceptions**

Let us suppose the site preparer *agent may fail* because of (1) missing materials or (2) bad weather.

# The building house example



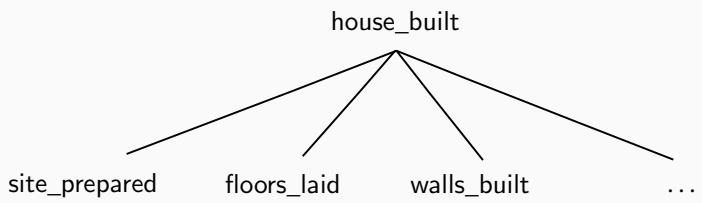

- R(*site preparer*, site_prepared) $\in$ **R**
- A(*site preparer*, *manager*, $\top$, site_prepared) $\in$ **A**

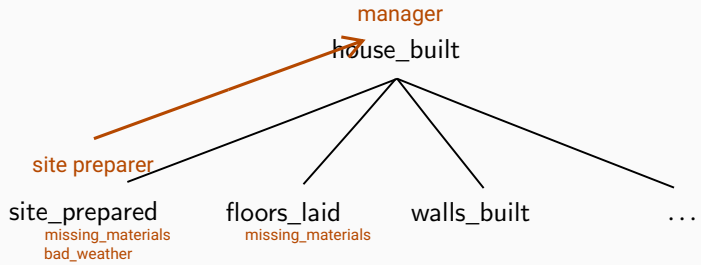
**Abnormal situations/exceptions**

Let us suppose the site preparer *agent may fail* because of (1) missing materials or (2) bad weather.

- *Site preparer* is *touched* by the two exceptional events *missing_material* and *bad_weather*.
- Thus (*site_prepared*, *missing_material*) $\in \mathcal{F}$ and (*site_prepared*, *bad_weather*) $\in \mathcal{F}$.

# The building house example




- R(*site preparer*, site_prepared) $\in$ **R**
- A(*site preparer*, *manager*, $\top$, site_prepared) $\in$ **A**

**Abnormal situations/exceptions**

Let us suppose the site preparer *agent may fail* because of (1) missing materials or (2) bad weather.

- *Site preparer* is *touched* by the two exceptional events *missing_material* and *bad_weather*.
- Thus (*site_prepared*, *missing_material*) $\in \mathcal{F}$ and (*site_prepared*, *bad_weather*) $\in \mathcal{F}$.

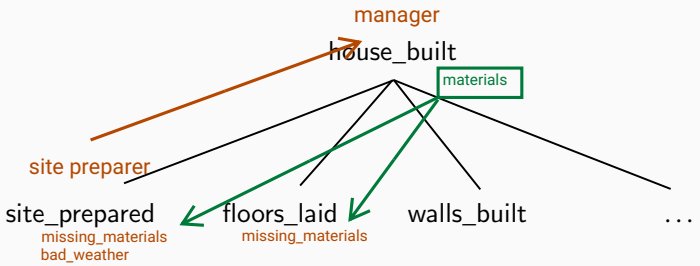
$[\mathbf{R} \rightsquigarrow \mathbf{A}]_{\mathcal{F}}$ characterizes what kinds of exceptional events should be reported and to who.

# The building house example

- Extending the functional decomposition by **enriching goal specifications** with the list of the respective relevant exceptional events that could thwart goal achievement.
- The responsible agent will be asked to **report** either the successful achievement or the **exception causing the failure**.

```
<functional-specification>

 <scheme id="build_house_sch">

  <goal id="house_built">
   <plan operator="sequence">
     <goal id="site_prepared" ttf="20 minutes">
       <exceptions>
         <exception id="missing_material" />
         <exception id="bad_weather" />
       </exceptions>
     </goal>
     <goal id="floors_laid" ttf="25 minutes">
       <exceptions>
         <exception id="bad_weather" />
         ...
       </exceptions>
     </goal>
     <goal id="walls_built" ttf="40 minutes" />
     ...
   </plan>
   <catch>
     <goal id="weather_emergency" handles="bad_weather">

       <plan operator="..."> ... </plan>
     </goal>
     <goal id="materials" handles="missing_material">
       <plan operator="sequence">
         <goal id="materials_got" ttf="10 minutes" />
          ...
       </plan>
     </goal>
    </catch>
  </goal>

    ...

 </scheme>

</functional-specification>
```

# The building house example




- The specification includes which **handlers** will be activated to tackle the *abnormal situations* (*exception handlers*).
- Handlers are modelled as goals to be achieved in alternative to the failed one (the goal of the agent who receives the exception).

```
1  <functional-specification>
2
3   <scheme id="build_house_sch">
4
5    <goal id="house_built">
6     <plan operator="sequence">
7       <goal id="site_prepared" ttf="20 minutes">
8         <exceptions>
9           <exception id="missing_material" />
10          <exception id="bad_weather" />
11        </exceptions>
12      </goal>
13      <goal id="floors_laid" ttf="25 minutes">
14        <exceptions>
15          <exception id="bad_weather" />
16          ...
17        </exceptions>
18      </goal>
19      <goal id="walls_built" ttf="40 minutes" />
20      ...
21    </plan>
22    <catch>
23      <goal id="weather_emergency" handles="bad_weather">
24
25        <plan operator="..."> ... </plan>
26      </goal>
27      <goal id="materials" handles="missing_material">
28        <plan operator="sequence">
29          <goal id="materials_got" ttf="10 minutes" />
30            ...
31        </plan>
32      </goal>
33    </catch>
34   </goal>
35
36     ...
37
38  </scheme>
39
40  </functional-specification>
```

# Conclusions

- Robustness in software systems is "the ability of a software to keep an 'acceptable' behavior [...] in spite of exceptional or unforeseen execution conditions (such as the unavailability of system resources, communication failures, invalid or stressful inputs, etc.)."
- Accountability is a **non-functional requirement** of a software system, that has a positive impact on **system robustness**, since it captures an **infrastructure** for analysing the organization's performance and take action if deemed necessary
- Beyond exceptions, accountability is an enabler for organization **adaptation**, both in **structure** and in **strategies**

# References

Baldoni, M., Baroglio, C., Boissier, O., May, K. M., Micalizio, R., and Tedeschi, S. (2018).
**Accountability and responsibility in agent organizations.**
In *PRIMA 2018: Principles and Practice of Multi-Agent Systems, 21st International Conference*, volume 11224 of *Lecture Notes in Computer Science*, pages 261–278. Springer.

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

# References ii

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