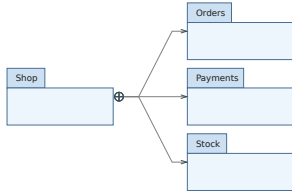


URDAD Cheat Sheet

The notation

Each view of a URDAD model shows one concern, and each is drawn here as the modelling tool draws it, from a small model of a shop. The terms link to the [glossary](#), and the [method](#) says how the views are used: a service, its requirements, who addresses each, and the process, repeated one level down.

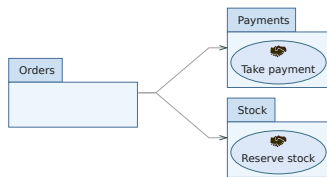
Scope View



What a [responsibility domain](#) is made of.

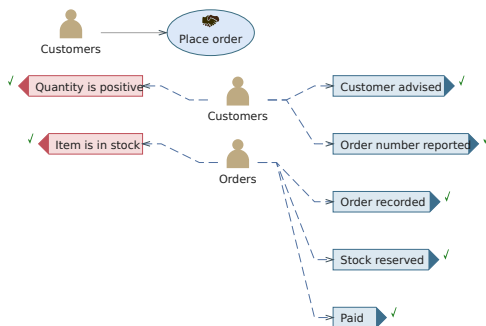
- **Box with a tab:** a responsibility domain, named on its tab.
- **Line from a $\oplus$:** containment: the domains it is divided into.

Dependencies View



The domains whose services a domain's processes request, and those services.

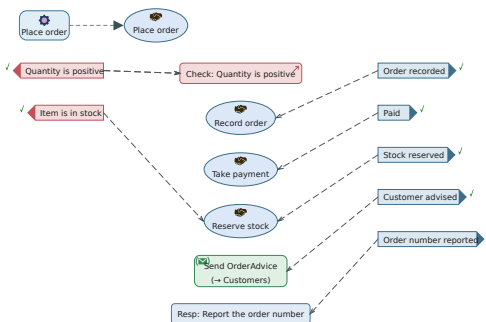
Functional Requirements View



What a [service contract](#) requires and guarantees.

- **Ellipse with a handshake:** the service contract.
- **Figure:** a [stakeholder](#): a domain with an interest. Above the contract, with a solid arrow to it, a domain the contract is published to, which may request it.
- **Red flag pointing left:** a [pre-condition](#): what must hold for the service to be provided. When it does not, the service refuses by raising its [exception](#).
- **Blue flag pointing right:** a [post-condition](#): what holds once the service is provided.
- **Dashed line:** the stakeholder who requires that condition.

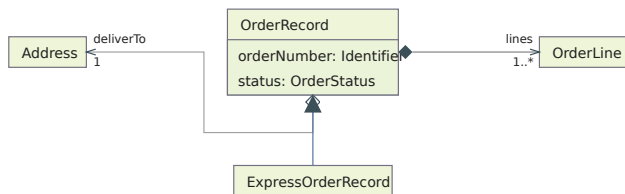
Responsibility Allocation View



Who addresses each requirement.

- **Box with a gear:** the [process](#), realizing the contract (dashed arrow with a closed head).
- **Dashed arrow:** from a requirement to the step that addresses it: a request of a lower-level service (an ellipse), a check, a message sent, or the response.

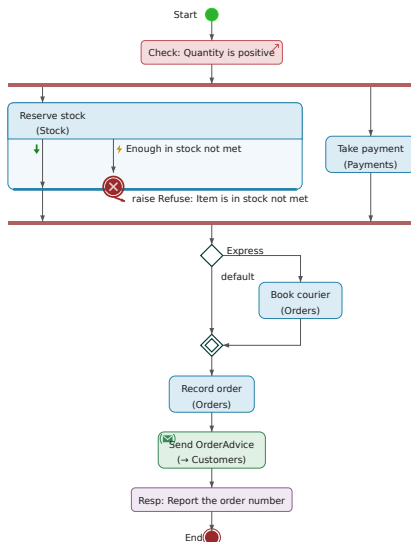
Data Structure View



A **data structure**, its fields and relationships.

- **Filled diamond:** composition: the part belongs to the whole.
- **Open diamond:** aggregation: the part is shared.
- **Open arrow alone:** association: a reference.
- **Closed triangle:** specialization: the structure below has every field of the one it points to, and more.
- **Numbers:** multiplicity: 1, 0..1, 1..*, ...

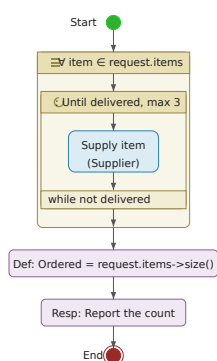
Process View



How the process orchestrates its steps, from start (green) to end (red ring).

- **Check (red box):** tests a pre-condition, and refuses if it does not hold.
- **Blue box:** a **service request**: the contract requested, and the domain providing it in parentheses.
- **Green arrow ↓:** the request's success path.
- **Yellow bolt ⚡:** an exception arm: the refusal named, and what is done about it.
- **Red circle with ×:** raise an exception: this service refuses.
- **Blue double diamond:** where a request's arms rejoin.
- **Two heavy bars:** **concurrent flows**: every lane runs, and all are joined.
- **Diamond:** **alternate flows**: the first conditional flow, in their order, whose condition holds is taken, else the default; the double diamond rejoins them.
- **Green box with ☒:** send a message, to the recipient after →.
- **Resp (purple box):** populate the response.

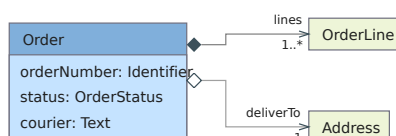
Loops and variables



Repeating steps.

- **Frame headed $\forall x \in xs$:** for each: its body runs once per element of the collection, one after another ($\equiv$) or all at once ($||$).
- **Frame headed $\forall$:** while: its body repeats while the condition in its footer holds, at most as often as the header says.
- **Def (purple box):** define a variable: name a value once, for the steps after it.

Entity View



An **entity**: a business object with an identity, with every field its views hold.