Please try to identify the modelling elements needed, to express the following story in a qualitative model

In the case of an accident at a nuclear power station, radioactive isotopes are emitted to the environment, where the amount of nuclear particles and level of nuclear radiation increases. For humans either direct contact with radiation, e.g. at the power plant itself or via uptake with food increases the probability of getting ill. The amount of different isotopes is increasing within different regions of the human body, increasing the risk of different types of illness.

Below, a summary of the characteristics of each learning space is presented:

A ***concept map*** (sometimes also referred to as an entity-relation graph) is a graphical representation that consists of two primitives: nodes and arcs. Nodes reflect important concepts, while arcs show the relationships between those concepts. From a qualitative reasoning point of view, concept maps are less interesting because they do not capture dynamics.
However, having this learning space is useful from a learning point of view, as it is the root from which more complex knowledge representations emerge.

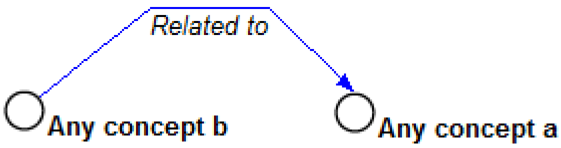
The learning space ***basic causal*** *model* focuses on quantities, how they change and how this change influences other quantities to change. Quantities represent behaviour. They are connected to entities, the structural units in the model. Simulation at this learning space means to calculate for each quantity one of the following options (for its derivative): decrease, steady, increase, ambiguous (because of opposing influences), or unknown (because of missing information).

The learning space ***basic causal model with state-graph*** augments the basic causal model learning space with the notion of a quantity space, which can be assigned to one or more quantities. This feature introduces concepts such as state-graph, behaviour path, and value history.

The learning space ***causal differentiation*** takes all details defined for the basic causal model with state-graph and refines certain notions, particularly those related to causality. Different from the preceding learning space is that the notion of exogenous quantity behaviour is included in the default setting. Also included in the default setting is the idea of an agent. Positive and negative feedback loops are possible!

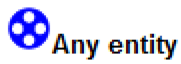
The learning space ***conditional knowledge*** is a refinement of the causal differentiation learning space. All representation details apply as they do for this preceding learning space. The main difference is the possibility to specify conditions under which specific set of details are true. In the preceding learning spaces all the expressions created by a learner are always true. However, some facts (e.g. the process of evaporation) only occur when certain conditions are satisfied. The conditional knowledge learning space addresses situations like these.

The learning space *generic and reusable knowledge* reflects the highest and most complex learning space of modelling. The main difference with the other learning space is the focus on 're-usable' knowledge. That is, to capture essential details in a context-independent manor as much as possible, or otherwise to explicate the conditions under which the knowledge is applicable. At this learning space the notion of types and hierarchy become important. Also the idea of learners creating their own models by re-using (partial) solutions stored in a repository now viable.



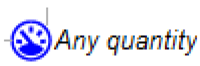
Concepts linked by relations

Idea: capture and structure relevant information.



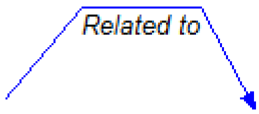
Entity

Entities are the physical objects or abstract concepts that constitute the system. Their relevant properties are represented as quantities.



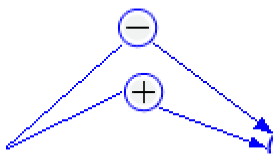
Quantity

Quantities assigned to entities are the dynamic features of a system holding information on magnitude and direction of change.



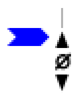
Configuration

Configurations are links between entities that are used to hierarchically or causally structure the system from a conceptual point of view.



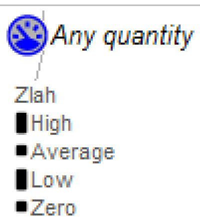
Causal relation

Causal relations describe positive or negative causal relationships between two quantities.



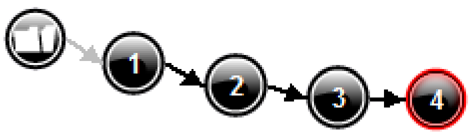
Derivative of change

The derivative of change holds information on direction of change (e.g. increase, decrease, steady).



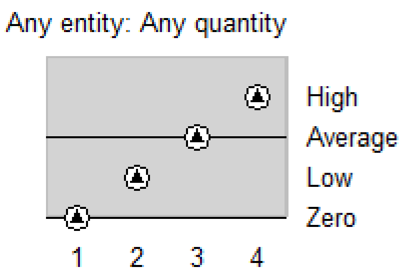
Quantity space

Quantity spaces represent changeable features of entities and contain an ordered set of system states and allow for simulations creating a state-graph with value history of system behaviour.



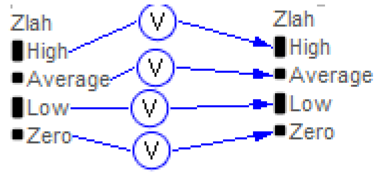
Simulation state graph

A state graph visualizes the succession of system states during a simulation.



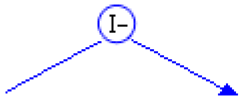
Simulation value history

Derived from a state graph, the value history shows the values quantities go through in a simulation.



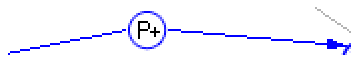
V-correspondences

V-correspondence between two magnitudes / quantity values providing simultaneity: if values of two quantities correspond, then when one occurs, the other one also occurs.



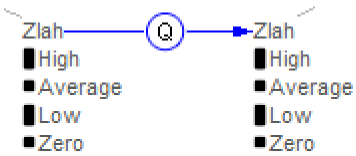
Direct influence

Direct influences are directed relations between two quantities, and are either positive or negative. Direct influences are the cause of change within a model, and are therefore said to model processes. They are used to add or subtract continuously from state variables.



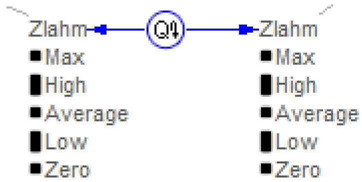
Proportionalities

Qualitative proportionalities are directed relations between two quantities and can also either be positive or negative.. They propagate the effects of a process from one quantity to another (like a correlation). For this reason, they are also referred to as indirect influences.



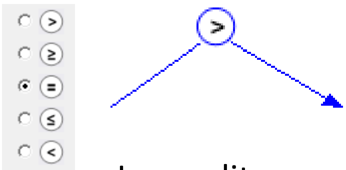
Directed Q-correspondence

Directed Q-correspondence is a correspondence between two quantity spaces providing simultaneity: if two quantity spaces correspond directed, their values change simultaneously during a simulation (if one value is high, the other one is also high). It`s used to reduce the complexity of the simulation.



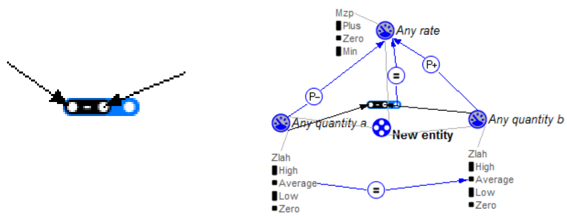
Inverse Q-correspondence

Inverse Q-correspondence is a correspondence between two quantity spaces providing simultaneity: if two quantity spaces correspond inversely, their values change inversely during a simulation (one goes from zero to high, the other one from high to zero). It`s used to reduce the complexity of the simulation.



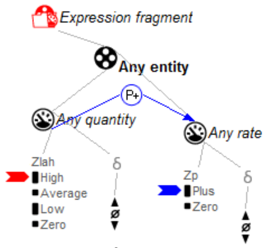
Inequality

Inequalities (<, ≤, =, ≥, >) specify an ordinal relation between two quantities i.e. that one item is different from (or equal to) the other item; e.g. allows to say when which rate is bigger/smaller to reduce when two rates are active on one state variable, ambiguity.



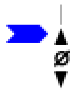
Calculus

The- calculus can be used to substract two quantity spaces to create a positive or negative rate.



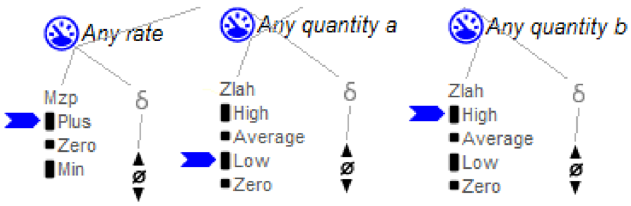
Condition

A process becomes only active, when a certain limit in the system is reached (red=condition, blue is consequence). This knowledge is specified in conditional expression fragments.



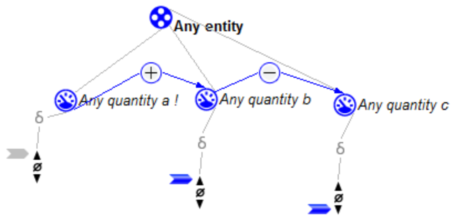
Value assignments

The value assignment describing the direction of change starts the simulation (e.g. increase, decrease, steady).



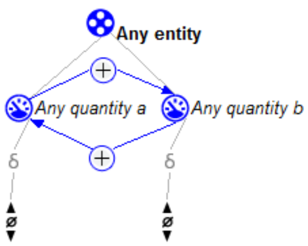
Value assignments

The value assignments to quantity spaces describe the starting values of a system before change takes place.



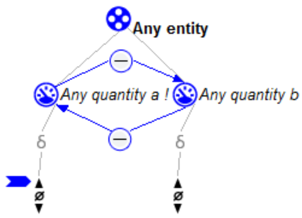
Simulated directions of change

After a starting value has been assigned to describe the direction of change, based on the causal relations, the directions of change of the other quantities are calculated.



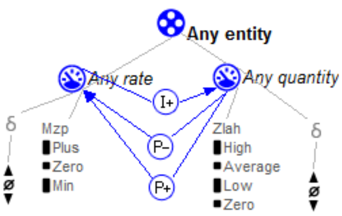
Positive feedback loop

If one quantity has a positive influence on another quantity, there could be only a positive feedback on the original quantity increasing this positive influence.



Negative feedback loop

If one quantity has a negative influence on another quantity, there could be only a negative feedback on the original quantity increasing this negativee influence.



Positive & negative feedback loop

If a rate has a positive influence on another quantity, there could be a negative and positive feedback on the rate increasing or decreasing its influence.

Modelling elements identification table

Nr	Learning Space	Introduced ingredients
1	Concept map:	• Entities • Configurations
2	Causal model:	• Entities • Configurations • Quantities o Value assignments -> direction of change (Derivatives) • Causal relationships (+ and –) • Simulation of directions of change, only shows increase/decrease/steady
3	Causal model with state graph:	• Entities • Configurations • Quantities • Quantity spaces o Value assignments -> direction of change (Derivatives) o Value assignments -> start value within quantity space • Causal relationships (+ and –) • Correspondences • Simulation has a state graph and value history
4	Causal differentiation:	• Entities • Configurations • Quantities • Quantity spaces o Value assignments -> direction of change (Derivatives) o Value assignments -> start value within quantity space • Causal relationships o Influences (=rates which add and substract) o Proportionalities (=correlations between state variables) • Correspondences • Inequalities (<, ≤, =, ≥, >) • Calculus (+ and –) • *Agents (no influence on simulations)* • *Assumptions (no influence on simulations)* • Simulation has a state graph and value history
5	Conditional knowledge:	• Entities • Configurations • Quantities • Quantity spaces • Value assignments -> direction of change (Derivatives) • Value assignments -> start value within quantity space • Causal relationships o Influences (=rates which add and substract) o Proportionalities (=correlations between state variables) • Correspondences • Inequalities (<, ≤, =, ≥, >) • Calculus (+ and –) • Conditional expressions • Simulation has a state graph and value history
6	Generic and reusable:	• All elements of above available • Different modelling approach – compositional modelling • Entity hierarchy • Model fragments • Scenarios that recall model fragments • Assumptions, that activate specific model fragments to be active within a scenario • Only needed to show that a certain behaviour is only true under certain circumstances, in a specific environment -> this can be implemented using assumptions in model fragments • Only build process model fragments • Simulation needs a scenario • Simulation has a state graph and value history

Decide, in which LS you will build your model given the elements you identified

<table>
<tr><td colspan="2">Evaluation:</td><td>Evaluators:</td></tr>
<tr><td colspan="2">Name of student:</td><td>Date:</td></tr>
<tr><td colspan="2">Which Learning Space you are going to use, to build the model? (LS 1-6)</td><td></td></tr>
<tr><td colspan="2">Entities – what are the main entities of the system you are looking at?</td><td></td></tr>
<tr><td colspan="2">Configurations – how are the entities linked?</td><td></td></tr>
<tr><td colspan="2">Quantities – what are the most important quantitative features in your system?</td><td></td></tr>
<tr><td rowspan="2">Quantities with their quantity spaces needed – decide on your quantity spaces, are there important landmarks/ relevant/critical values</td><td>State variables (linked by P`s or positive/negative causal relations), describe the state of a quantity</td><td></td></tr>
<tr><td>Rates – add or substract constantly from a state variable – what is the most relevant rate in your system?</td><td></td></tr>
<tr><td colspan="2">Is there a feedback loop to be implemented? How are you going to do that?</td><td></td></tr>
<tr><td colspan="2">Conditional knowledge – a specific process only takes place if a certain value is reached</td><td></td></tr>
</table>