

FEEDBACK LOOPS

Feedback loops are the **stories** that tell what happens in the system. If you plan to map your system, use feedback loops as building blocks of your system map.

A system is driven by forces, and those cannot exist in isolation. Each force has a cause and effect which are interrelated. The forces that drive the system are always tied together in feedback loops.

Building a loop

1. Start by picking a parameter that you think is important. A parameter is a **noun** which can **increase** or **decrease**. This can refer to a measurable quantity or a quality.
2. Work downstream from that parameter. What does it cause? Look for the 'effect' among the other parameters you have. Once you have found it, ask yourself what does it cause, and find another 'effect' parameter. Go on finding **downstream effects** among your parameters: ultimately, if it is a loop, you will have a parameter which causes the original, starting parameter.
3. As you are developing your causal loop, document how each parameter affects the downstream parameter. Look at the loop in fig.1:
 - if the increase of parameter A is creating an increase in parameter B, write '+' at the beginning of the arrow, and '+' at its end. The parameters grow in the **same direction**, meaning that if A grows, B grows as well and if A decreases, B decreases as well. Indicate it by writing **(S)** near the arrow.
 - If an increase in factor B determines a decrease in factor C, write + at the beginning of the arrow and - at its end. The parameters grow in the **opposite direction**: if B increases, C decreases and, symmetrically, if B decreases, C increases. Indicate it by writing **(O)** near the arrow.
4. Once you have formed a promising loop, give it a brief yet descriptive name.

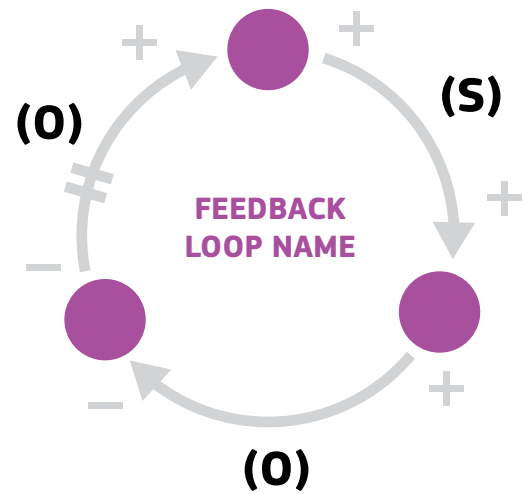


fig.1 Structure of a generic feedback loop.

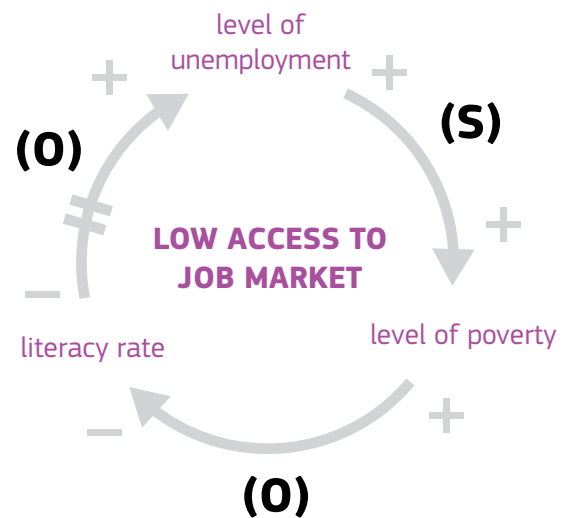
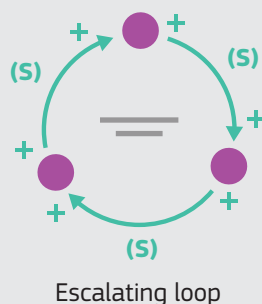
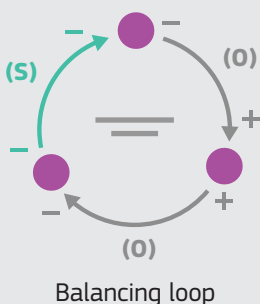


fig.2 An example of story told through a feedback loop.

LEGEND



- parameter
- (S) growth in the same direction
- (O) growth in the opposite direction
- + "if/then the parameter increases..."
- "if/then the parameter decreases..."
- ++ delay