



A Domain-Specific Language to Process Causal Loop Diagrams with R

OR2019 – Dresden – September 2019

Adrian Stämpfli - Institute of Modeling and Simulation, University of Applied Sciences St. Gallen

adrian.staempfli@fhsg.ch

Abstract

Status Quo:

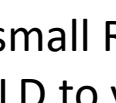
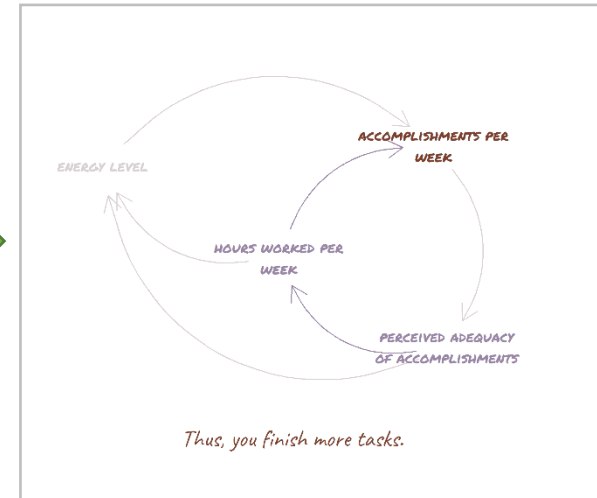
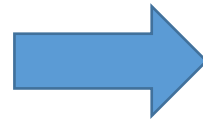
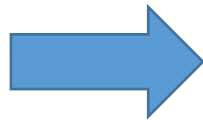
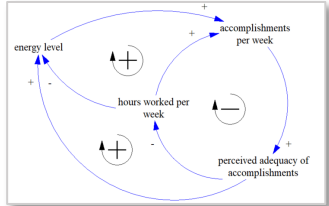
- Causal Loop Diagrams (CLDs): A flexible and valuable tool used in strategic decision-making and management
- The crucial dissemination of CLDs and the possible learnings beyond the project-team is challenging

Proposed solution:

- A Domain-Specific Language (DSL) that allows modeling experts to generate visual representations of CLDs that
 - i. Omit the most complicated aspects of CLDs;
 - ii. Compensate for this information loss, and
 - iii. Strive to lower the barriers to learning, while addressing a broader target audience

Targeted Use Case

Causal Loop Diagram



R-Package with Fuctinonality
for CLD-Visualization (the DSL)



1 Create your CLD using a System Dynamics-Tool (i.e. Vensim)

2 Export the CLD as an *.mdl-File

3 Write a small R Script to explain the CLD to your Audience using new DSL Commands

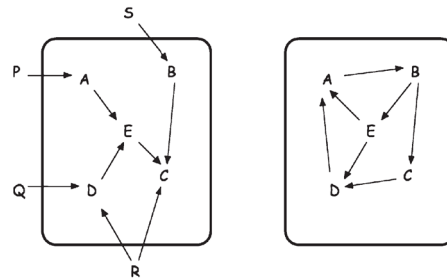
4 Get a new Representation of your CLD which is easier to grasp



Introduction – System Dynamics and CLDs

System Dynamics (SD)

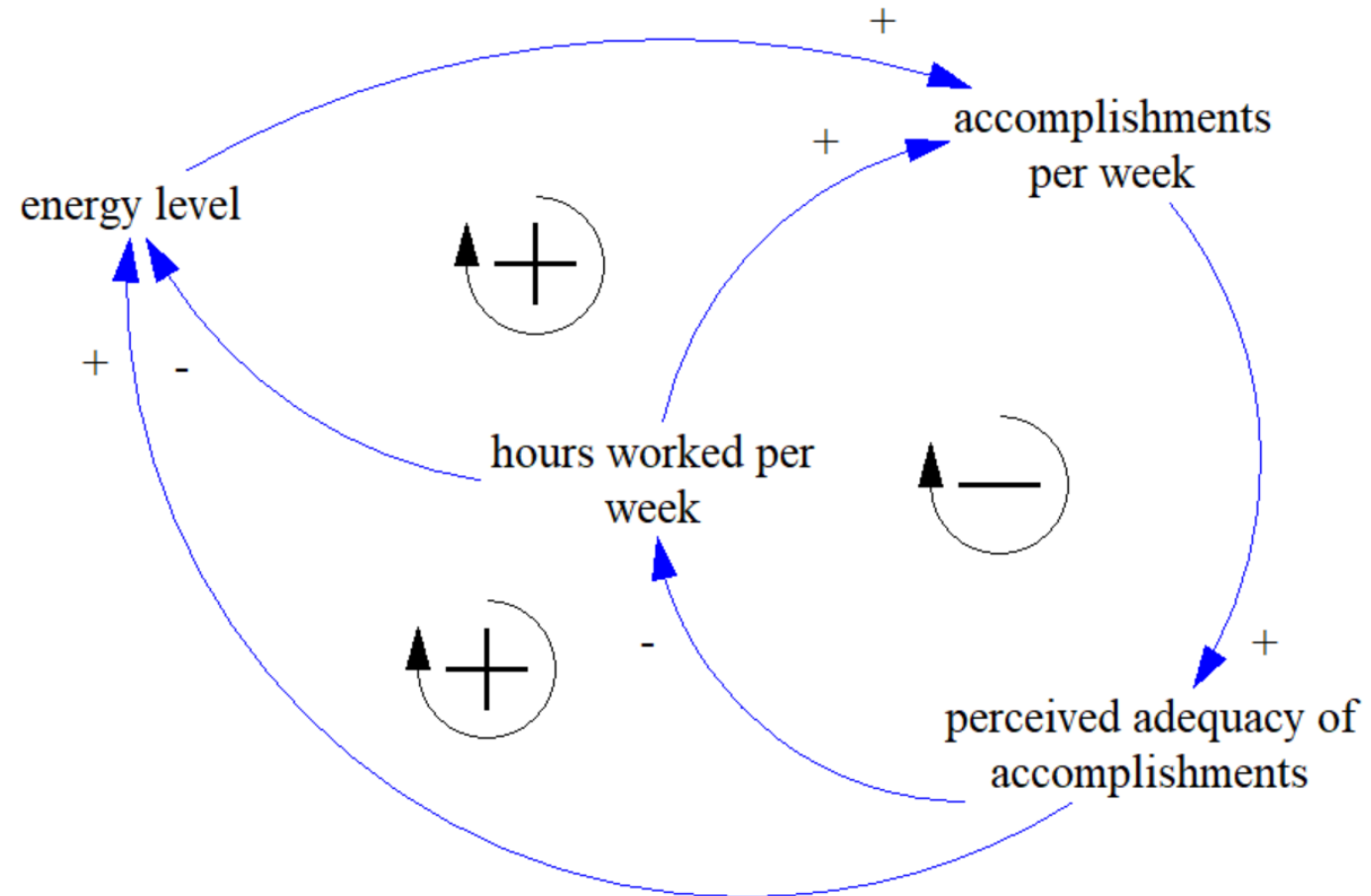
- Method for **modeling and simulation of complex systems** that adapts control theory to a broader set of problems (Forrester1961)
- Two key elements differentiate SD from other methods in Operations Research:
 - SD models *generate dynamics endogenously*. Many classical SD models show how flawed internal policies of industries or cities generate decay without external limiting factors (Richardson 2011).



- SD *makes mental models explicit* by modeling them as CLDs. Making the models explicit is the basis for a deeper understanding of a messy situation, for revising mental models, for allowing double-loop learning to occur and for taking strategic decisions (Torres2017¹, Lane1992¹, Vennix1999, Paich1993²)

¹) published in "European Journal of Operational Research", ²) published in "Management Science"

Causal Loop Diagrams – What they are



Legend

	Variables (Vertices)
	Causal Links (Edges)
	Direct Proportionality
	Inverse Proportionality
	Balancing Loop
	Reinforcing Loop

Slightly simplified "Worker burnout model" by Jack B. Homers (Homer 1985)

Usage of Causal Loop Diagrams

CLDs are used

- i. to structure complex problems;
- ii. to explore complex decision-making situations in participatory modeling processes;
- iii. to foster learning among stakeholders involved in the modeling process;
- iv. as a basis for simulation models, and
- v. to communicate results of simulation studies

General Findings:

- System Dynamics is best applied to relatively complex and unstructured problems where endogenous dynamics generate (unwanted) dynamics (Hovmand 2014, Lane 1999, Vennix 1999)
- Customer involvement is crucial in such problems: Confidence in models and simulation results is necessary for implementation to happen (Black 2013, Wolstenholme 1999, Hovmand 2014)
- While CLDs are valuable *in* such projects, their dissemination *out* of the project is unlikely (Wolstenholme 1999, Hovmand 2014)
- Relevant stakeholders (senior decision-makers) generally don't have the knowledge to interpret CLDs (Wolstenholme 1999). They don't like being “taught” (Wolstenholme 1999).

So the question is: How can we strengthen the dissemination of CLDs outside the project team?



Methodology

R

- open-source programming language and software environment designed for statistical computing, data science, and graphics (Ihaka & Gentleman, 1996)
- R is a very flexible language
 - Computing on the language is straightforward in R (Wickham 2015, Mailund 2018)
 - The combination of first-class environments, lexical scoping, non-standard evaluation, and meta-programming make R especially well suited to support the creation of embedded Domain Specific Languages (DSLs) (Wickham, 2015)
- We use R at the Institute of Modeling and Simulation
 - sim911: A package published on CRAN to analyze rescue-service based data
 - Communication: Shiny Apps, R Markdown

Domain-Specific Languages

Domain-specific language (noun): a computer programming language of limited expressiveness focused on a particular domain (Fowler, 2011).



Solution

Functionality

- **IMPORT** a CLD from Vensim
- **LINK** group CLD elements (using existing Causal Chains)
- **DESCRIBE** groups of CLD elements with textual descriptions
- **PLOT** the resulting CLD

Grammar – Sentences

```
DSL_SENTENCE ::= DSL_EXPRESSION  
              | DSL_EXPRESSION %>% PLOT
```

```
DSL_EXPRESSION ::= IMPORT  
                | DSL_EXPRESSION %>% LINK  
                | DSL_EXPRESSION %>% DESCRIBE
```

Grammar – Causal Chains

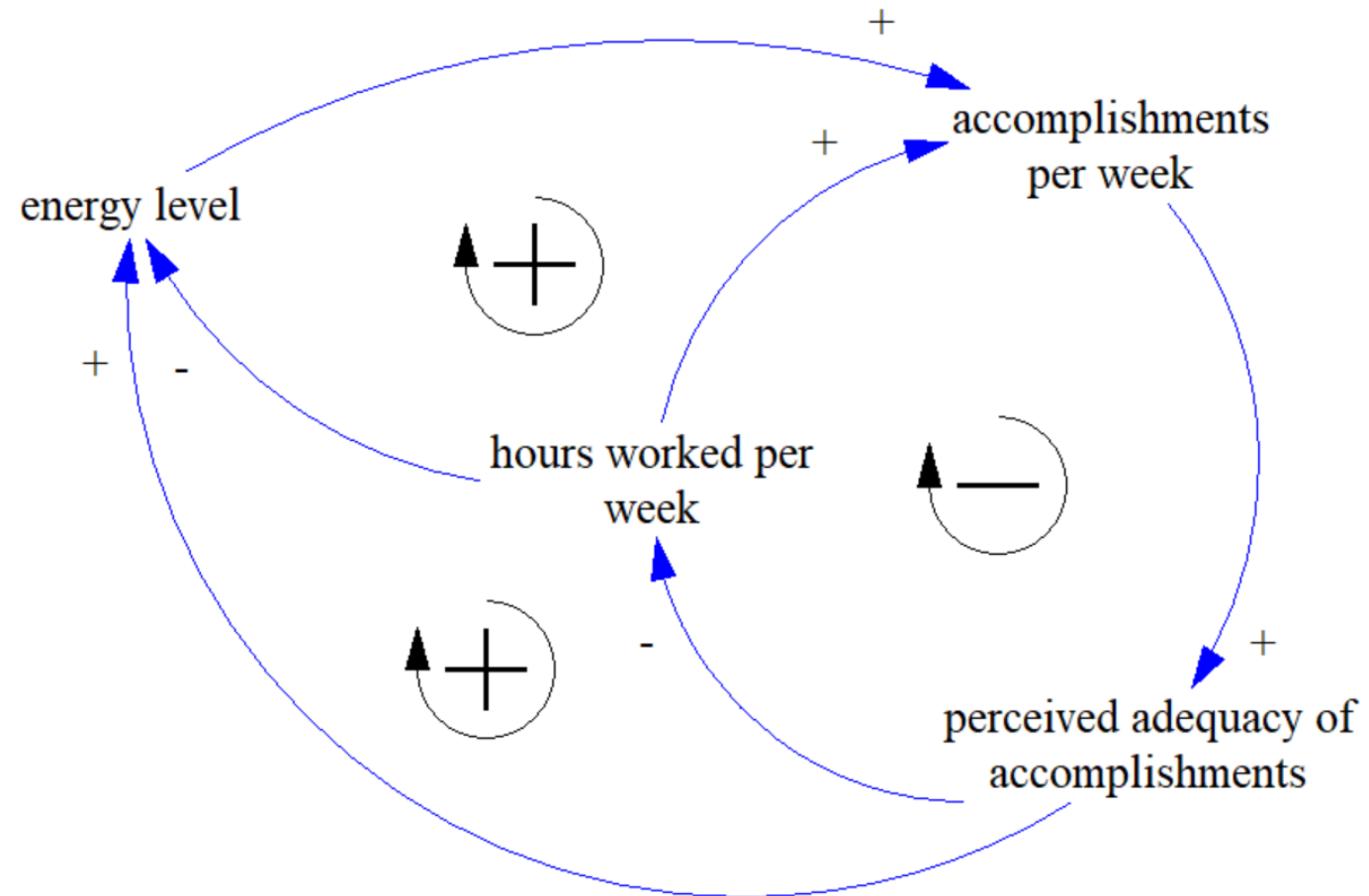
```
LINK_EXPRESSION ::= CAUSAL_CHAIN  
                | CAUSAL_CHAIN , LINK_EXPRESSION
```

```
CAUSAL_CHAIN ::= VARIABLE  
             | CAUSAL_CHAIN %->% VARIABLE
```



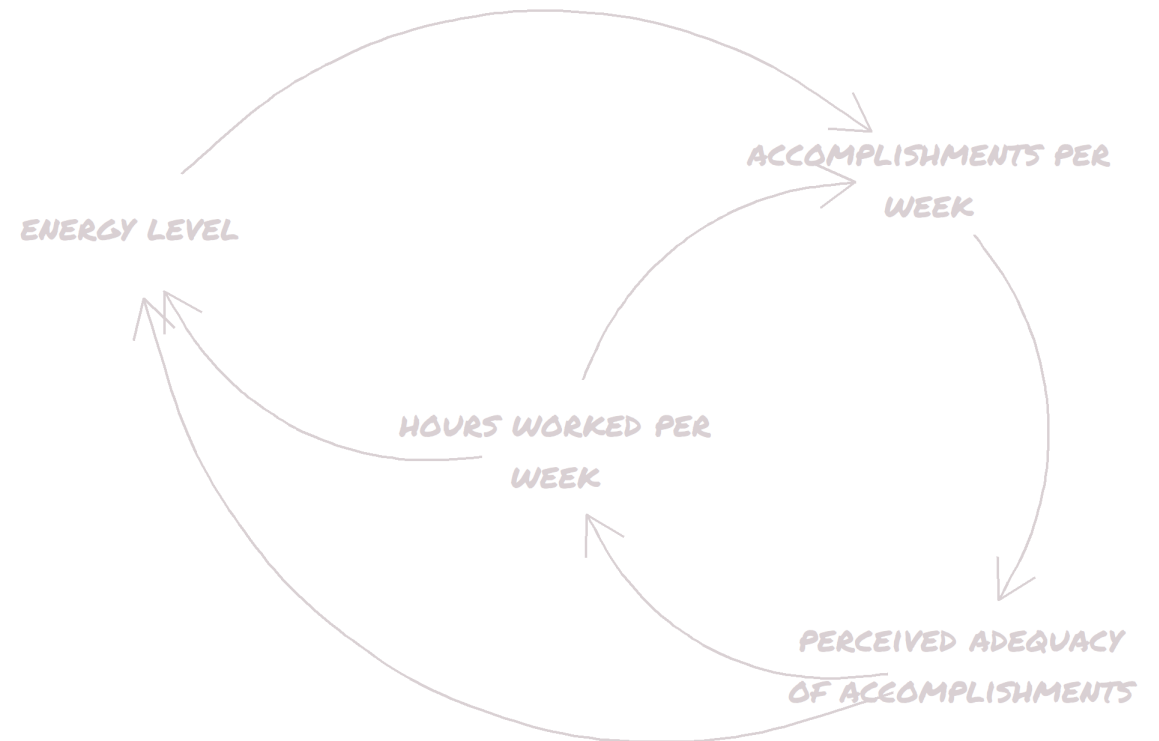
The DSL in Action

The original CLD «burnout model» (Homer 1985)



Default plot

`cloud %>% plot()`

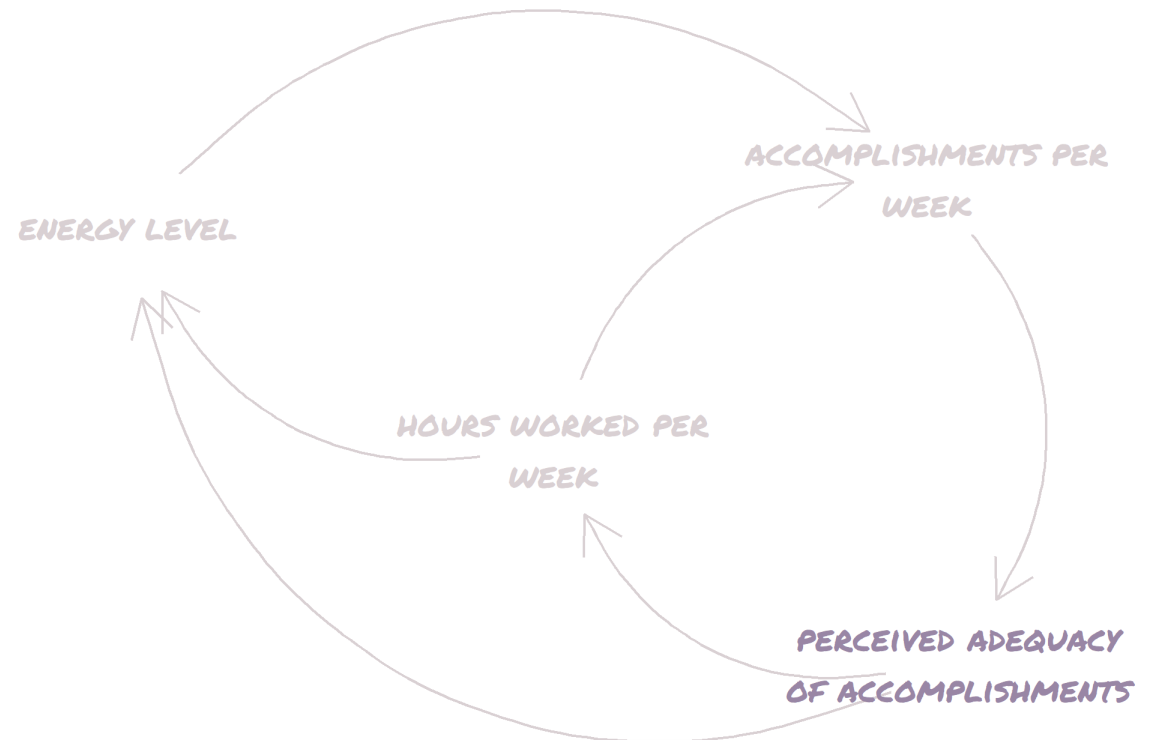


Highlighting a single variable

```
cmd %>%
```

```
  link(`perceived adequacy`) %>%
```

```
  plot()
```



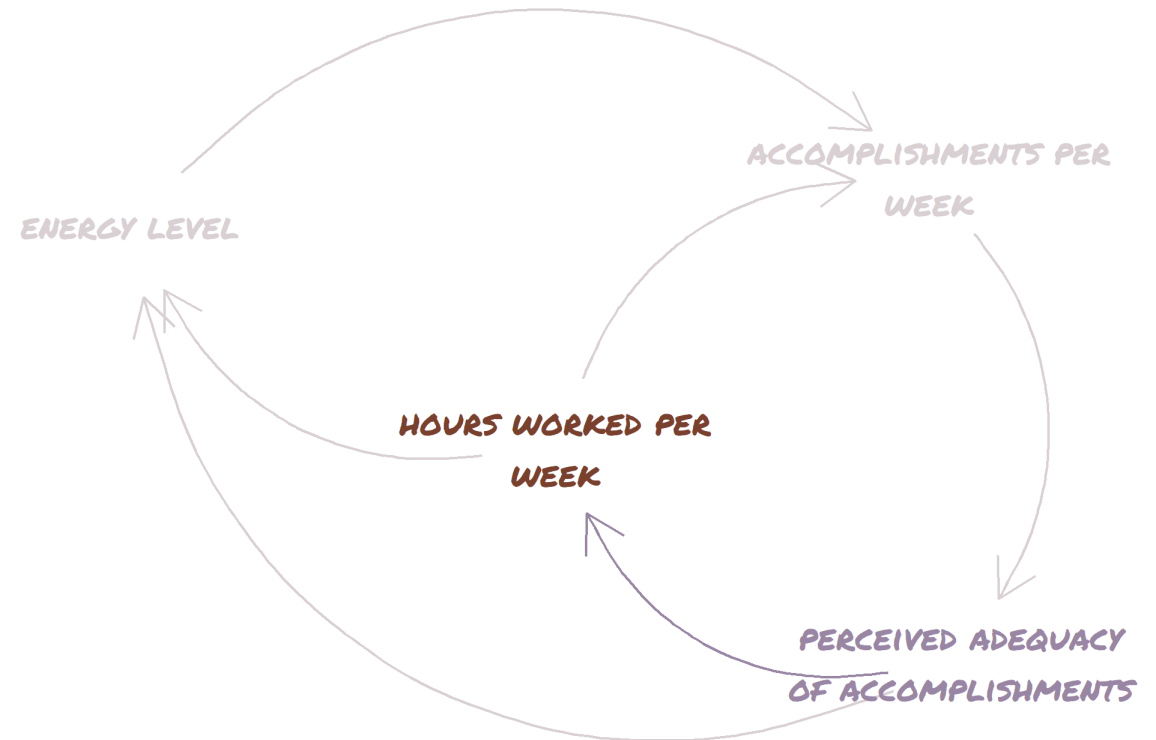
Define a causal chain

```
clId %>%
```

```
  link(`perceived adequacy` %>% `hours worked`) %>%
```

```
  link(`hours worked`) %>%
```

```
  plot()
```



Add a textual explanation

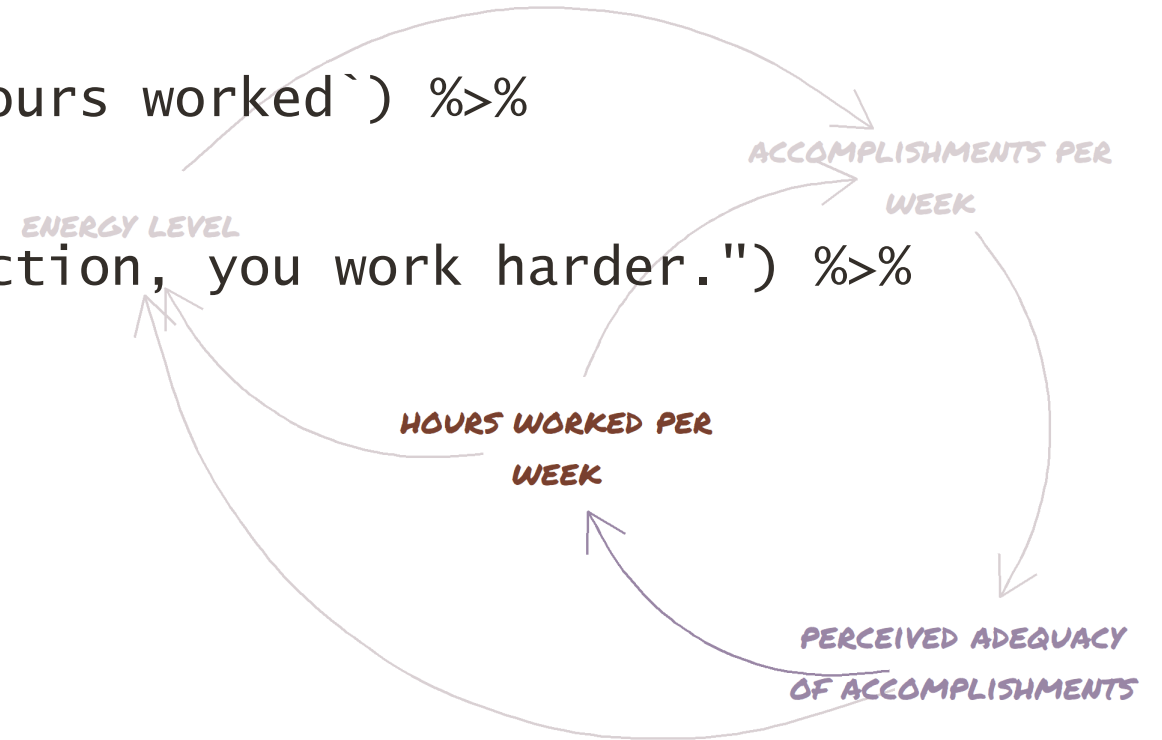
```
cld %>%
```

```
  link(`perceived adequacy` %>% `hours worked`) %>%
```

```
  link(`hours worked`) %>%
```

```
  describe(type = "text", "As a reaction, you work harder.") %>%
```

```
  plot()
```



As a reaction, you work harder.

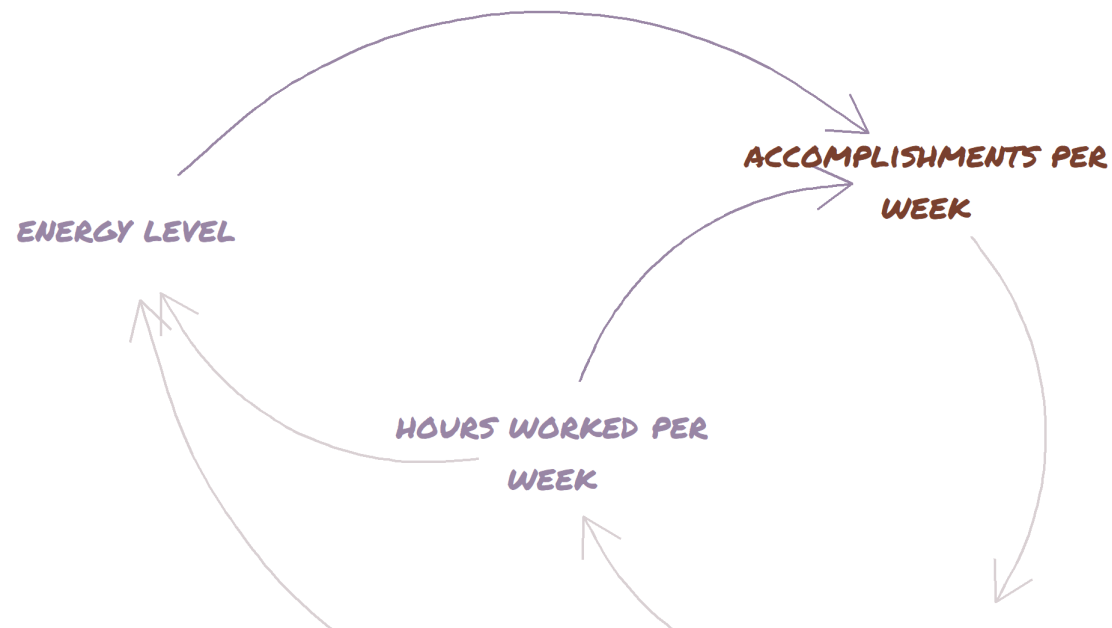
Two causal chains

```
cld %>%
```

```
  link(`energy` %->% `accomplishments per week`,  
        `hours worked` %->% `accomplishments per week`) %>%
```

```
  link(`accomplishments per week`) %>%
```

```
  plot()
```





Conclusions

Conclusions: A New Approach to Communicate CLDs

- DSL allows us to explain CLDs **bit-by-bit**
 - To ensure that the CLD's **circular structure is not concealed**, the whole model is always kept visible (greyed out)
 - Highlighting certain elements helps to **break the CLD into understandable pieces**
- **Compensate for the information loss*** by providing additional descriptions
- **Reach a broader target audience** through a less technical look

* caused by the missing link- & loop-polarity signs

Conclusions: Embedded DSL in R

- To implement the solution in the form of an embedded DSL in R proves valuable
- Thanks to the DSL approach, we can write short, simple, and elegant code, which in turn provides for excellent prototyping possibilities
- R's properties allowed us to find surprisingly simple notations, grammars, and suitable plotting possibilities.

Conclusions: Applications

In numerous customer projects, the DSL turned out to be a very valuable tool:

- i. to develop a common problem understanding;
 - ii. to communicate that understanding to stakeholders beyond the project team;
 - iii. to foster strategic decision-making.
- A particular appealing application of the developed DSL is a project funded by 'Innosuisse - Swiss Innovation Agency' in the field of policy design for elderly care.

Resources

- R-Code is hosted at: <https://github.com/ims-fhs/cld>
- Models from the «Work-Life Balance 4.0» project (German):
<https://www.fhsg.ch/de/forschung-dienstleistungen/institute-zentren/institut-fuer-soziale-arbeit-und-raeume/integration-und-arbeit/work-life-balance-40/wirkungszusammenhaenge/>
- The “Burnout Model” App (German): <https://fhsg.shinyapps.io/burnout/>
- Information on «Who Cares», the on-going project in elderly care:
<https://www.fhsg.ch/de/forschung-dienstleistungen/institute-zentren/institut-fuer-modellbildung-simulation/care-system-design/verbesserte-planung-der-langzeitpflege/>

If you have any questions, please feel free to contact me: adrian.staempfli@fhsg.ch



Discussion



Thank you!