Design for a dynamic decision experiment platform to capture and analyze mental models.

June 2021

Martin FG. Schaffernicht [1], Joel Fuentes [2], Miguel López-Astorga [3], Ramón Castillo and Cristian Rojas-Barahona [4]

1. Introduction

This short note describes the general design of a computational platform for the research proposal "How to achieve sustainable growth in a feedback-rich system? The development of mental models and decision rules of naïve decision-makers in a dynamic task experiment." (Chilean National Scientific Fund *Fondecyt* ID# 447625). It introduces the outline of how we will capture data by designing the action space of participants such as to derive raw data for representing their mental models, their decision policies, the decisions, and their outcomes. The note first introduces the platform's architecture. Then the general procedure for the experiment sessions is outlined. Then section 4 discusses how the interactive causal diagram and the constructors capture raw data and how it is converted I not several categorical measures.

2. The computational platform

The platform has two layers: *user interface* and *model, procedures, and database*. Data is mainly captured through the specially developed user interface, allowing participants to receive explanations, to interact with the simulator running the model with the target system, to view and analyze what happens through an interactive causal diagram which they develop themselves and to interactively formulate their policies. Figure 3 displays the general architecture. The interface layer comprises a series of control panels which participants can use in any phase from *briefing* through *policy application*. During *policy construction*, the questions asked by the (virtual) manager also appear in these panels.

[1] Facultad de Economía y Negocios, Universidad de Talca, Avenida Lircay sn, 3460000 Talca (Chile). martin@utalca.cl

[2] Facultad de Ciencias Empresariales, Universidad del Bio Bio (Chile)

[3] Instituto de Estudios Humanísticos Abate Molina, Universidad de Talca (Chile)

[4] Both Facultad de Psicología, Universidad de Talca (Chile)

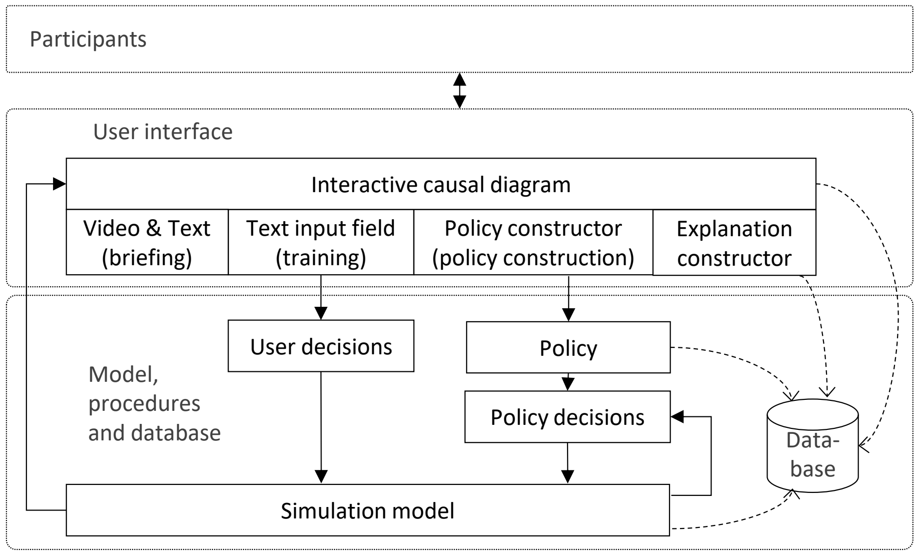

Figure 1: The architecture of the experiment

The *interactive causal diagram* is an essential component of the user interface. The variables and the causal links participants believe to be relevant can be inserted from a pop-up menu. Each variable appears with its name and a minimized graph which can be enlarged. The graph displays the variable's behavior so far and participants must sketch the value they expect to see in the next season (in response to their decision):

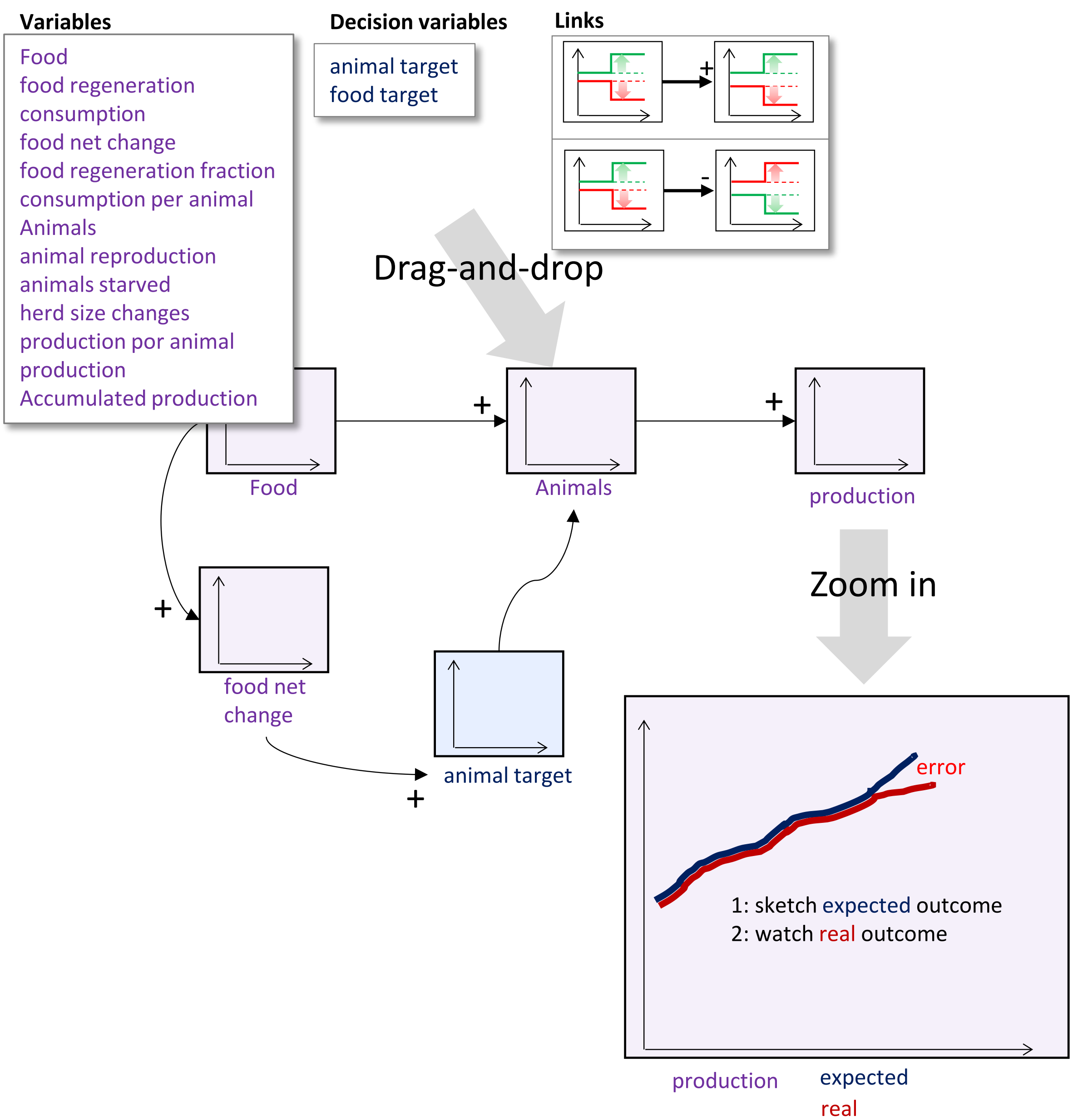

Figure 5: The interactive causal diagram

The second input component is the *policy constructor* (Figure 6), in which participants develop the rules allowing to set a value for the decision variables they use (*animal target* and possibly the *food target*). Variables accounted for can be inserted from a pop-up menu containing all variables linked to the used decision variables in the causal diagram (this enforces consistency), and all necessary operators can be inserted by click.

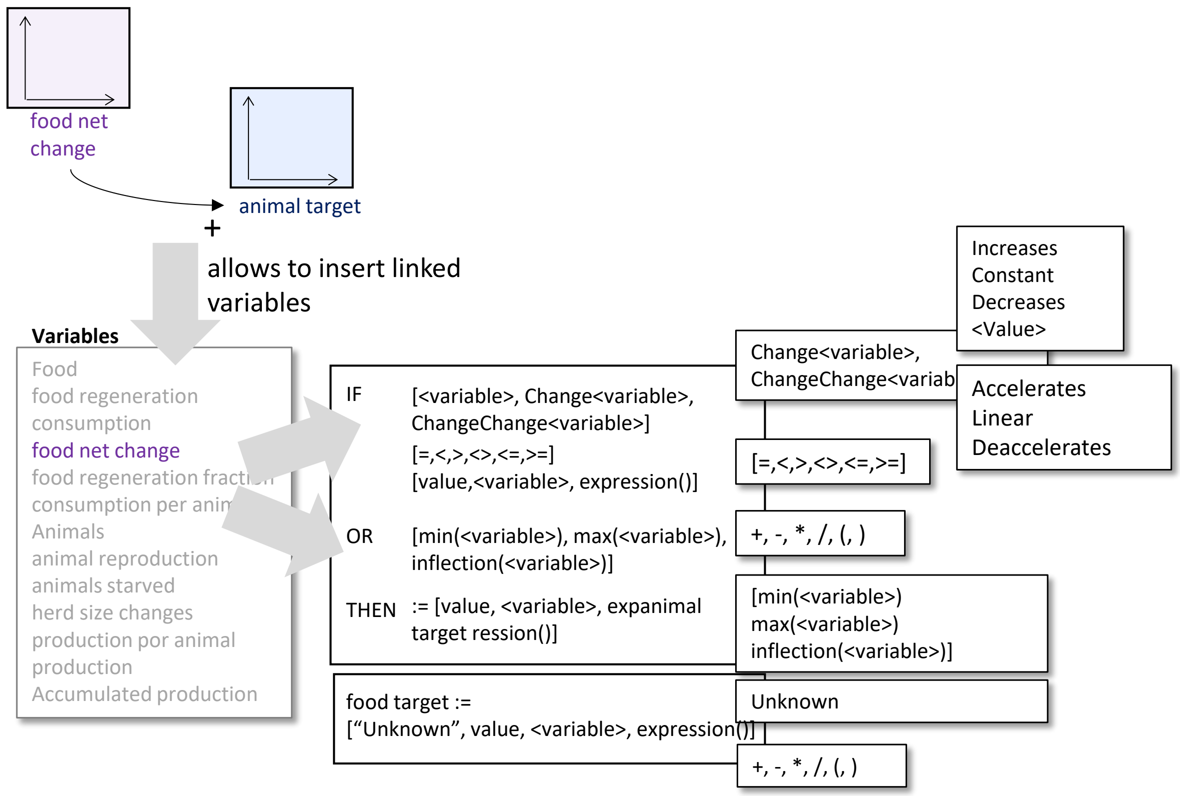

Figure 6: The policy constructor and explanations constructor interface

The causal diagrams and the policies are stored by version because participants can change them during the experimental session.

3. Procedure

Before starting, participants must read and sign up the informed consent approved by the Scientific Ethics Committee of our institution. Each participant will follow a sequence of phases: pretest; briefing; training; policy construction; policy application; exit recommendation.

In the *pretest*, participants respond to comprehension questions designed to elicit their systems thinking skills and possible prior knowledge.

For the experiment, the *target system* and the task are like a game with the following roles. The *owner* is a fictitious person recruiting a new *herd manager*. New managers are *trainees* for one game before becoming *herd manager*. The computational platform plays all roles except the participant’s one.

The *briefing* provides a description in video and written document: the *owner* explains the situation and the task to the new *herd manager*. This includes the fictitious report from the previous manager, where the non-linear relationship between *food* and *food* *regeneration* is implicit in the data contained. These explanations use a non-technical language, without directly providing the precise structure of the computational model and its equations. They remain available during the entire experiment, together with a factsheet.

The *training* phase lasts for one complete game, during which participants play the role of *trainee*. In each season, the *trainee* can directly input a decision (without specifying an articulated policy). Before each season, the *manager* asks: "What effects do you expect?" and "Why do you believe that this will work?" - two questions intended to prompt the *trainee* to observe and reflect on the decisions and their outcomes. The answers are typed in, using an edit field similar to the *policy constructor*'s features.

In the *policy construction* phase, participants design their own *policy*. The platform provides an interface segment in which participants can interactively construct decision rules (production rules) and a conditional format ("If … then do …"). The conditions can refer to any combination of the available variables; for each variable, the condition can specify values, first differences (increases, decreases) and second order differences (acceleration, deceleration). Logical operators (AND, OR, XOR) can combine conditional segments. Why constructing the policy, participants can use the currently defined rules as a prototype and make the simulator carry them out: this allows to test if the current set of decision rules steers behaviors and the performance which the participants esteem satisfying. No limit is imposed on the number of such test trials, but the number of trials is recorded for later analysis. In a final construction step, participants define the final set of "abort" rules in the form "If … then abort". If in any season of the *policy application* phase, such a rule is triggered, the simulation will stop, and the session returns to the *policy construction* phase: it has become obvious that the current *policy* is not satisfying in at least one known circumstance. Policies and the underlying mental models are stored to the database for each version (each time a trial is stopped or aborted, a new version begins), together with the start and stop time and the number of seasons carried out in the respective trial.

The *policy application* phase is the execution of the *policy* through the seasons of the decision task, and participants are now in the role of *herd manager*. The *assistant* is now a piece of software interpreting the *decision rules* and providing input to the simulation model. In this phase, participants will not interact with the model, but the behavior and performance steered by this *policy* will be displayed for them at the end of the last season. They can then choose either to switch back to the *policy construction* phase and try to change the policy or choose to keep the current results and finish their game.

The *exit recommendation* prompts participants to indicate which policy they recommend to their successor in their role as *herd manager* and to explain why this would be a good recommendation. Participants can recommend their own *policy* or any modified version of it.

4. Data acquisition and analysis

4.1 Raw data and construction of categorical measures

We will collect the data required in laboratory experiments, exposing participants to the dynamic task through "microworld" - an on-line computational platform carrying out the simulation of the underlying computational model representing the *target system* and allowing participants to interact with this model (Gonzalez, Fakhari, & Busemeyer, 2017). A number of features will be recorded in the platform's database as raw data for analysis: (a) the performance, (b) the policies, (c) the MMP and (d) the MMDS will be used to evaluate each participant. As part of the raw data, the database will contain the values for each model variable (per season/iteration): Food, consumption per animal, production per animal, food regeneration, Animals, production, consumption, animal reproduction, Accumulated production, food net change, animals starved, food regeneration fraction, herd size changes.

Error feedback implementation. Error feedback is applied in studies **S2** and **S3** in the conditions C2 and C4. Its implementation takes advantage of the fact that in the training phase, participants are prompted to sketch the intended outcome of each decision before they execute it. After each season, if there is a gap between intended and observed outcomes, the manager asks: "why did it go wrong?". Participants have the "explanation constructor", a window analogous to the "policy constructor", to type in an explanation that uses the variables and links of the causal map. This window contains the question "may you have overlooked something?" with a navigation link to the causal map, as a prompt to revise and reconsider the causal map, and then return to the explanation constructor.

Prompting full deployment of MMPs. Prompting for full deployment is applied in studies **S2** and **S3** in the conditions C3 and C4. It is also computationally aided. The causal map contains the variables and causal links attributed to the situation. Each causal link joins a pair of variables, and together with the polarity (positive or negative), a pair of conditionals is derived from the causal link.

For positive polarity:

- "<variable 1> increases → <variable 2> will have higher values that it would have had."
- "<variable 1> decreases → <variable 2> will have lower values that it would have had."

For negative polarity:

- "<variable 1> increases → <variable 2> will have lower values that it would have had."
- "<variable 1> decreases → <variable 2> will have higher values that it would have had."

The possibilities will be derived from the corresponding conditionals, and each possibility is fed into a question on-screen: "Could it happen that <possibility>?", prompting one of the following answers: Yes, Maybe, Not likely, No, I cannot tell, Impossible to know.

4.2 Relevant features

(a) Regarding performance, the *accumulated production*, as compared to the optimal value yields a percentage which we call relative accumulated production or RAP, a quantitative measure of goal achievement. RAP is used to assign participants to one of the following categories:

1. Optimum: RAP >= 95%,
2. High: 75% <= RAP < 95%,
3. Intermediate: 50% <= RAP < 75%,
4. Low: 25% <= RAP < 50%,
5. Poor: 25% <= RAP < 50%.

The trajectory of *food* over is used to assess if sustainability has been achieved (*food* must not have a decreasing trend) as a categorical measure.

(b) Regarding the policies expressed by participants, one question is if a *food target* has been set (categorical); if so, the next question is if the value was correct (categorical); if not, the third question is of the participant has used one of the variables *food*, *food net change* or *food regeneration* (categorical). The *animal target* is always set; however, the question is if it was set at the correct value (categorical); if not, the question is if one of the following variables was mentioned as input: *food net change* or *food gap*, which is the difference between *food target* and *food* if a *food target* is set (categorical). Policies will also be categorized according to the generic policies described above: 1, 2, 3 or "other".

(c) The mental models of possibilities (MMP) are reconstructed from the policy rules – which can be expressed as conditionals to derive implied possibilities, and their expected effects. The Fully Explicit Model, which contains all possibilities, is parsed to determine which are the relevant possibilities (possibilities which refer to something that can happen in the task). Analysis of the participants' data reveals which mental models have been deployed and which have not. The number of non-deployed but relevant MMP (MMP errors) as compared to the number of relevant MMP is RAMMP (relative adjustment of mental models of possibilities): the percentage of deployed relevant possibilities. Participant's RAMMP score in this measure is assigned to the following categories:

1. All: (RAMMP = 100%),
2. Most (75% <= RAMMP < 100%),
3. Many (50% <= RAMMP < 75%),
4. Few (25% <= RAMMP < 50%),
5. and None (0% <= RAMMP < 25%).

Some of the *target system*'s feature may have been recognized but not used in the reasoning. Therefore, the MMP will be parsed to detect if the following features have been detected or not: *food regeneration*, *food net change*, the *starvation* driver, the *delay* between *food regeneration* driver and herd size changes, and the non-linear relationship between *food* and *food regeneration*.

(d) The mental models of the dynamic system (MMDS) are stored in part in the database of the causal diagrams maintained by the participants, and in part extractible from the MMP: the conditionals mention variables and causal links, and unmentioned elements have evidently not been considered relevant by the participant. Variables or causal links which are not contained in the conditionals or in the causal diagram constitute MMDS errors. The feedback loops which have not been recognized are also MMDS errors. Five features are either recognized or not: *food regeneration*, *food net change*, the *starvation* driver, the *delay* between *food regeneration* driver and herd size changes, and the non-linear relationship between *food* and *food regeneration*.

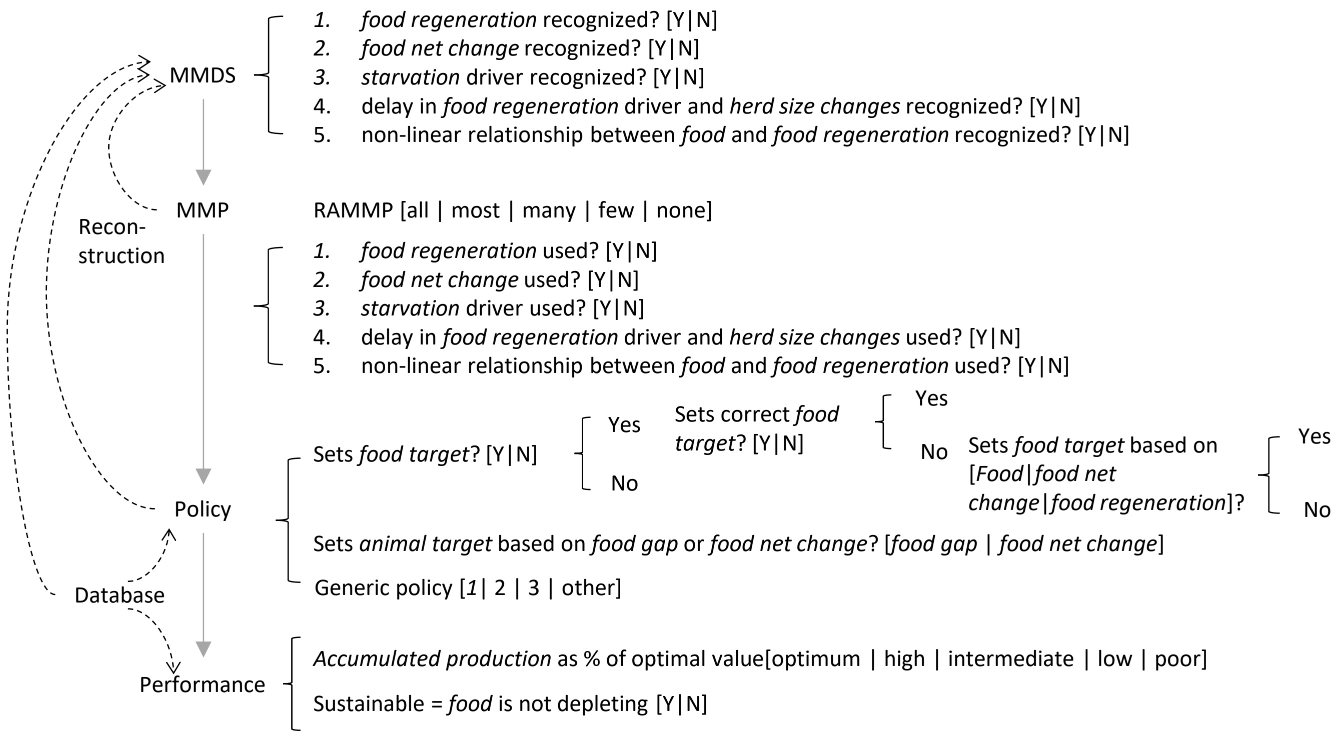

Figure 2: From data to analysis

4.3 Analysis

The individual data of each group's participants will be aggregated for study **S1** and each of the four conditions of studies **S2** and **S3**. This feeds into the statistical tests we will use to corroborate the project's hypotheses.