SchemaFit Timer as an Instrument

Graph, Schema and AI-assisted analysis in one training cycle

Serega Photoskiff

2026

Contents

Introduction **1**

1. A timer clicks; SchemaFit Timer models **2**

2. Why the graph is necessary **2**

3. Two levels of the model **3**
Personal layer 3
Workout layer 3

4. Schema: a language for complex workouts **3**

5. A common language and exchange format **4**

6. The stencil: merging graph and Schema **4**

7. Why intensity is often omitted **5**

8. Variable Schemas as reproducible experiments **5**

9. Real-time control **6**

10. The AI analysis file **6**

11. The complete cycle **6**

Introduction

SchemaFit Timer is more than an interval timer and more than a heart-rate chart. It connects visual planning, a formal description of load, heart-rate prediction, workout execution, measured heart rate and post-session analysis.

An ordinary timer manages time: it says when to work, when to recover and when to finish. SchemaFit Timer treats a workout as a cycle of **model, execution and validation**.

```
Graph + Schema + personal model
               v
     predicted heart rate
               v
          workout
               v
      measured heart rate
               v
```

```
      comparison and analysis
              v
     refined model and Schema
```

The system has two main ways to describe a workout. The graph provides a fast visual starting point and parameters for the physiological model. Schema provides a powerful language for complex load. Together they make it possible not only to run a workout, but also to understand how it was expected to unfold, how it actually unfolded and what should be refined next time.

1. A timer clicks; SchemaFit Timer models

At its core, an ordinary interval timer emits a sequence of signals: start work, stop work, start recovery, stop recovery. It may have a polished interface, sounds, presets and history, but it still answers one principal question: **when does the phase change?**

SchemaFit Timer works with a broader object:

- intended load;
- predicted heart-rate response;
- measured heart rate;
- deviation from the model;
- quality of the match;
- subsequent refinement of model parameters.

Even a basic heart-rate model needs more information than a clock. For a single interval atom it needs at least four dynamic parameters:

- work duration;
- recovery duration;
- work intensity;
- recovery intensity.

A repeated or variable session also needs its repetition and transformation rules.

To relate the prediction to a person rather than to an abstract athlete, the system also uses personal parameters: heart-rate range, zones, response and recovery behaviour, time constant `tau`, drift and other model settings.

That is the fundamental distinction. A conventional timer can execute a session without knowing intensity, zones or physiological response. A system that predicts heart rate and checks that prediction against reality requires both a **load model** and a **person model**.

2. Why the graph is necessary

A text Schema often specifies only the stable structure: durations, repeats, progressions or variability. Intensities may intentionally remain unspecified because they can be calibrated later against measured heart rate.

For example:

```
10/8*10#
```

defines work duration, recovery and repeats, but says nothing about how hard work or recovery should be. A conventional timer could still click through the session. A heart-rate model cannot yet describe it fully.

The graph is therefore not merely a visualisation. It is a visual environment for setting missing model parameters. Its controls can define work intensity, recovery intensity and the other values needed for prediction.

The graph also works as an alternative editor for a simple atom. With no text Schema, its settings may form a complete Schema such as:

```
40/20@8/4*6#
```

When text is present, the graph supplies missing values and presents the final merged model that the timer will execute.

3. Two levels of the model

SchemaFit Timer joins two layers.

Personal layer

This describes the athlete:

- resting and maximal heart rate;
- heart-rate zones;
- response and recovery parameters;
- `tau`;
- drift;
- other individual physiological settings.

Workout layer

This describes the particular session:

- work and recovery time;
- intensities;
- repeats;
- grouping and progression;
- tempo;
- controlled variability;
- other load parameters.

The result is a heart-rate model for **this workout performed by this person**.

The same external session can have a very different physiological cost for two athletes - or for the same athlete at two stages of preparation. A useful model must describe both the shape of the load and the person meeting it.

4. Schema: a language for complex workouts

Sliders are convenient for a simple atom. They cannot naturally express a complex sequence of atoms, ladders, waves, progressive changes, seeded random sessions, scoped blocks or adaptive tempo.

This is where Schema becomes essential. It describes not only durations and pauses, but the **shape of load over time**.

A simple Schema:

```
10/8*10#
```

A variable pull-up ladder:

```
[dw=2 dr=2 N=10]
(dw/dr@r1->[to:auto/auto@rN,delta:+dw/+dr@r+1]
->[to:auto/auto@r1,delta:-dw/-dr@r-1])*1`#
title=pull-up ladder
```

It rises from one repetition to `N`, then returns to one, scaling work and recovery with the target repetitions. A bank of sliders cannot express that structure clearly.

The graph and Schema are therefore complementary. The graph is a visual editor and parameter source for a simple model. Schema is a language for complex structure and controlled variability.

5. A common language and exchange format

Natural-language workout instructions are often ambiguous. “Do a pull-up ladder with equal recovery” may be interpreted differently by every reader. Pauses, transitions and counting conventions quietly change.

A Schema is formal, compact and executable. It can be sent to another person and materialised by the timer as the same intended structure.

This enables:

- personal and public Schema libraries;
- coach-to-athlete exchange;
- community collections;
- reproducible tests;
- research protocols;
- sharing by text or QR code.

Schema is not only a timer input format. It is a common language for exchanging precise training models.

6. The stencil: merging graph and Schema

The graph and text Schema form two layers of one model.

The lower graph layer is always complete and can produce a simple Schema:

```
40/20@8/4*6#
```

The user’s text Schema forms an upper stencil. Explicit values cover the corresponding values below; omitted values are holes through which the lower layer remains visible.

If the graph defines `40/20@8/4*6#` and the user enters only a work duration of `50`, the resulting model becomes:

```
50/20@8/4*6#
```

If the user enters:

```
10/8*10#
```

work, recovery and repeats come from the text; the missing intensities come from the graph:

```
10/8@8/4*10#
```

This gives the graph three roles:

1. editor of simple Schemas;
2. source of missing values for partial Schemas;
3. visual presenter of the merged model used by prediction and execution.

7. Why intensity is often omitted

Durations, structure and repetition targets are frequently more stable than intensity. Intensity may need to be calibrated for the person, movement, implement weight, preparation phase, fatigue and current condition.

Omitting intensity at first makes the graph a calibration instrument. After a session, the timer compares modelled and measured heart rate. The athlete can tune work and recovery intensities until the model better describes the observed response.

Once suitable values have survived repeated checks, they can be written into the Schema and saved as a personally calibrated preset.

```
structure first
      v
initial intensity estimate
      v
measured session
      v
model comparison
      v
calibrated portable Schema
```

Leaving intensity open is therefore not a weakness. It supports individual calibration.

8. Variable Schemas as reproducible experiments

Schema can express tempo ramps, waves, progressions, random scenarios, repeatable randomness through `seed`, and reference-based work. The Schema becomes more than a timer instruction: it becomes a reproducible experiment on load shape and athlete response.

Repeating the same Schema - or the same seeded random materialisation - allows comparison of physiological cost:

- lower heart rate at the same load;
- reduced drift;
- faster recovery;

- later entry into severe zones;
- a cheaper finishing effort;
- higher tempo at the same heart-rate cost.

For an advanced athlete, this supports tactical comparison. For an ordinary athlete, it makes progress visible and helps prevent uncontrolled escalation.

9. Real-time control

The final merged Schema is not only drawn before the session. It guides the live workout.

The timer moves through work and recovery phases, receives heart-rate data from a monitor and compares measured heart rate with the modelled curve. A meaningful deviation can be reported while it is happening.

This may reveal that:

- the load is heavier or lighter than expected;
- recovery is slower than predicted;
- the athlete’s present response differs from their model;
- the data quality is insufficient for a strong conclusion.

The model is therefore a live reference, not decorative pre-session artwork.

10. The AI analysis file

Formal comparison can produce correlation, RMSE, lag, overlap, data-quality estimates, plausibility checks for `tau`, recovery characteristics and other metrics. These are valuable, but several related metrics can be difficult to interpret together.

SchemaFit Timer can therefore produce a structured AI analysis file containing:

- model parameters;
- realised workout parameters;
- comparison metrics;
- reliability rules;
- heart-rate series when required.

An AI system can use that file to explain what happened, how much confidence the result deserves, what might be changed in the next session, and whether a mismatch is more likely to come from the model, physiology or data quality.

The formal computation remains primary. AI does not replace it; AI translates the calculated evidence into an intelligible explanation and supports a continuing training dialogue.

11. The complete cycle

The value of SchemaFit Timer lies in the connection of its parts:

- the graph provides a fast start, visual control and an initial model;
- Schema describes a workout of arbitrary structural complexity;
- Schema also acts as a portable exchange format;
- the stencil combines text structure with graph parameters;
- the personal model connects the session to one athlete;
- modelled and measured heart rate support calibration;
- the timer controls execution and monitors deviation in real time;

- the AI file turns formal results into useful interpretation.

The compact formula is:

```
personal model + Schema + graph
-> predicted heart rate
-> execution
-> measured heart rate
-> comparison
-> AI-assisted interpretation
-> refined model and next Schema
```

SchemaFit Timer is therefore not just an interval timer, visualiser or notation system. It is an integrated instrument for planning, predicting, performing, analysing and personalising a workout.