UNIVERSITÉ DE GENÈVE

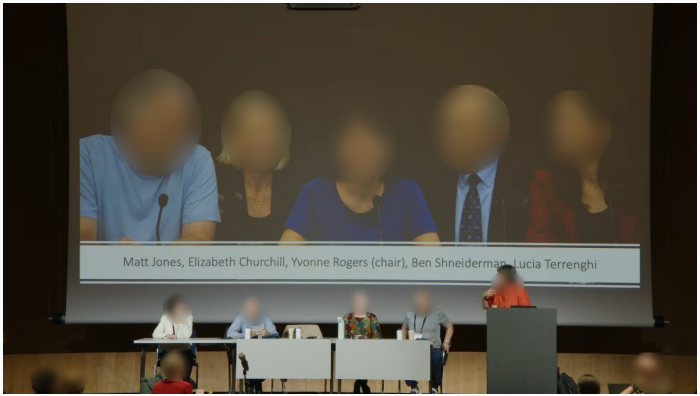
FNSNF
SWISS NATIONAL SCIENCE FOUNDATION

Why Constrained Interfaces are the Right Default for LLMs in Programming Education

Alexandre De Masi

VL/HCC 2026 • Paderborn • 2 October 2026

TECFA, Technologies for Training and Learning
Faculty of Psychology and Educational Sciences, University of Geneva



Chair: Yvonne Rogers • Elizabeth Churchill, Matt Jones, Lucia Terrenghi, Ben Shneiderman

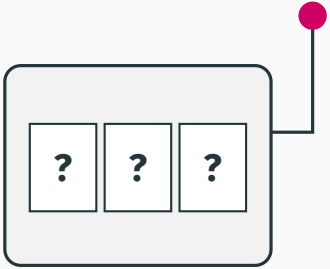
Free-form chat fails the students it should help

- The gap widens
- Delegation lowers the post-test
- The weakest students lean on it most

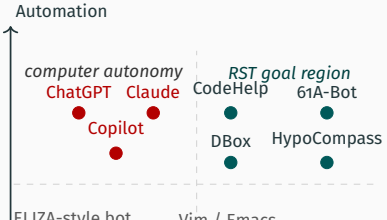
Faster today, no gain at one week.

Prather et al., ICER ’24 • Kazemitabaar et al., Koli Calling ’23 and CHI ’23 • Margulieux et al., ITiCSE ’24

Control means predicting the response



free to type, unable to predict



Six recurring constraint patterns

No direct answer
hints and pseudo-code, never the solution
CodeAid, CodeHelp

Hint ladder
subgoal, next step, near-code
One Step at a Time

Socratic
the model asks, the learner answers
CodeTutor, SocraticAI

Co-construction
the learner builds a structure first
DBox

Channel cap
one shot, inside the autograder
61A-Bot

Role inversion
the LLM is the buggy student
HypoCompass

Each comes with its own rationale. Nothing says which constraint to pick for which failure, or why.

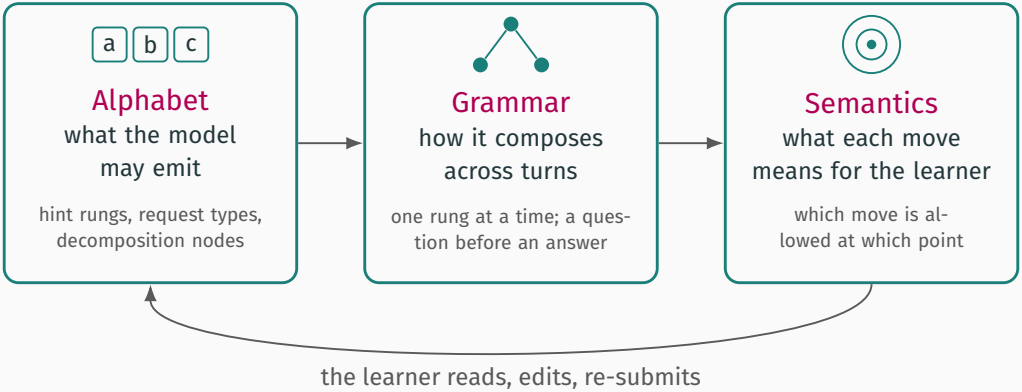
Each fix is an interface rule

Golden Rule	Constraint pattern	How free-form chat breaks it
1 Consistency of the response shape	hint ladder	a new response shape for every prompt
3 Informative feedback	co-construction	confident prose, right or wrong
5 Prevent errors	no direct answer, channel cap	the open box invites over-asking and copy-paste
7 Keep users in control	Socratic, role inversion	free to type, unable to predict
8 Reduce memory load	all six	a cookbook of prompts to memorise

Rules 2, 4 and 6 (universal usability, closure, easy reversal) hold for chat and constrained interfaces alike, so they do not separate them.

Golden Rules: Shneiderman, *Designing the User Interface*, 1987; wording from the 6th ed., 2016

A constrained interface is a language



Miss one of the three and you get chat with a system prompt.

A default, not a ban

- Loosen the constraints as expertise grows: advanced learners, open exploration, debugging, project work
- The evidence is still thin: two small within-subjects studies
- Next: interaction traces as designed objects, randomised classroom trials, a benchmark for the pedagogical quality of LLM output

For a novice, control means predicting the response.
Until they can, the interface should constrain it.

Thank you

Alexandre De Masi
alexandre.demasi@unige.ch

ademasi.ch

The system behind this argument, next week at NordiCHI 2026:

Designing Constrained AI Assistance for Novice Programmers: Tensions and Hypotheses from a JupyterLab Hint System

with Chen Wang and Laurent Moccozet

Monday 5 October, 15:30–17:00 • Help for Learning session
Levón A202, University of Vaasa

Backup: isn’t this learner-centred design?

In its lineage, yes: tools for learning should differ from tools for productivity.

What the paper adds:

- a criterion for which constraint answers which interface failure
- response-shape unpredictability as the failure specific to generative interfaces

Quintana, Krajcik and Soloway, ICLS 2013

Backup: why not a descriptive taxonomy?

A 10-dimension taxonomy of coding assistants (interface, inputs, outputs, autonomy, transparency, …) locates systems; it does not say which ones help novices learn.

DBox and an unconstrained chat tutor with the same model and materials sit in the same cell on the dimensions that apply. Only DBox improved learning (N=24, within-subjects). The difference is one constraint: feedback tied to a structure the learner built (Rule 3).

The descriptive view locates; the normative mapping identifies the responsible feature.

Lau and Guo, VL/HCC ’25 • DBox: Ma et al., CHI ’25

Backup: the evidence at its real size

Study	Design	Finding
Prather et al., ICER ’24	think-aloud, N=21, observational	gap widens
Kazemitabaar et al., Koli ’23	thematic analysis, N=33	delegation, lower post-test
Kazemitabaar et al., CHI ’23	controlled experiment	faster now, no gain at one week
Margulieux et al., ITiCSE ’24	mixed methods, self-report	lowest self-efficacy, most use
DBox, CHI ’25	within-subjects, N=24	learning gains over free-form
HypoCompass, AIED ’24	within-subjects, small	transfer to debugging

Backup: when is free-form chat the right call?

When the learner can already predict and steer the response:

- advanced learners
- open-ended exploration and project work
- debugging support, once the schema exists

The default loosens as the schema grows. The claim is about the default for novices.

Backup: isn’t the mapping post-hoc?

The mapping fails if

- a constraint pattern improves novice learning while satisfying none of the five rules, or
- a free-form feature satisfies all five and still harms learning.

DBox predates the paper, so the claim is explanatory: feedback tied to the learner’s tree (Rule 3) is the active ingredient. An ablation can test it: remove the tie, keep the model and the materials.

Backup: full mapping

Pattern	R1	R3	R5	R7	R8	Examples
No direct answer			●	○	○	CodeAid, CodeHelp
Hint ladder	●				○	One Step at a Time
Socratic	○			●	○	CodeTutor, SocraticAI
Co-construction		●			○	DBox
Channel cap			●		○	61A-Bot
Role inversion				●	○	HypoCompass

● primary design rationale ○ secondary or side-effect

Backup: counter-arguments

- Productivity gains are real. They are; throughput is not the goal of an introductory course.
- Bans hurt students without private access. A constrained institutional channel shifts the equity problem; on current evidence it is no worse, and plausibly better.
- Students will use ChatGPT anyway. Then the institutional channel has to win on learning; students prefer a structured scaffold and learn more from it.

Lau and Guo, ICER '23 • Hou et al., L@S '24 (CodeTailor)