STACK with state ⇒ *Stateful*

Matti Harjula

Aalto University

EAMS 2020
24th of June

Outline

1. Around 2012 we were spreading the joy of STACK to fields beyond maths and physics. People teaching industrial management needed optimisation related questions but what they really wanted were simulations and those were difficult.

1. Around 2012 we were spreading the joy of STACK to fields beyond maths and physics. People teaching industrial management needed optimisation related questions but what they really wanted were simulations and those were difficult.

2. After that we carried out various trials with JavaScript to implement state-based logic on the client side.

History of *Stateful* 1/2

1. Around 2012 we were spreading the joy of STACK to fields beyond maths and physics. People teaching industrial management needed optimisation related questions but what they really wanted were simulations and those were difficult.
2. After that we carried out various trials with JavaScript to implement state-based logic on the client side.
3. During 2015 a proof of concept of STACK with state was built, it was promising and was presented at EAMS 2016.

History of *Stateful* 1/2

1. Around 2012 we were spreading the joy of STACK to fields beyond maths and physics. People teaching industrial management needed optimisation related questions but what they really wanted were simulations and those were difficult.

2. After that we carried out various trials with JavaScript to implement state-based logic on the client side.

3. During 2015 a proof of concept of STACK with state was built, it was promising and was presented at EAMS 2016.

4. However, authoring questions for that proof of concept was difficult. So, an idea for separating state from STACK was hatched and the search for funding to implement it began.

5. In 2018 funding materialised, 10 months of development time funded for me to build it and some side projects to carry out authoring and usage testing. All part of a massive set of projects that the Ministry of Education and Culture of Finland accepted.

5. In 2018 funding materialised, 10 months of development time funded for me to build it and some side projects to carry out authoring and usage testing. All part of a massive set of projects that the Ministry of Education and Culture of Finland accepted.

6. Had to skip EAMS 2018 as coding started. However, I was very happy to see the Mathspace keynote by Mohamad Jebara, that essentially gave us the terminology for what we do: *Stateful* is all about "inner loop adaptation". I recommend watching that recording if you have not seen it

5. In 2018 funding materialised, 10 months of development time funded for me to build it and some side projects to carry out authoring and usage testing. All part of a massive set of projects that the Ministry of Education and Culture of Finland accepted.

6. Had to skip EAMS 2018 as coding started. However, I was very happy to see the Mathspace keynote by Mohamad Jebara, that essentially gave us the terminology for what we do: *Stateful* is all about "inner loop adaptation". I recommend watching that recording if you have not seen it

7. Summer 2019, major development goals ready and question authors started testing. Initial experiences were great and authors came up with unexpected patterns that led to new development.

History of *Stateful* 2/2

5. In 2018 funding materialised, 10 months of development time funded for me to build it and some side projects to carry out authoring and usage testing. All part of a massive set of projects that the Ministry of Education and Culture of Finland accepted.

6. Had to skip EAMS 2018 as coding started. However, I was very happy to see the Mathspace keynote by Mohamad Jebara, that essentially gave us the terminology for what we do: *Stateful* is all about "inner loop adaptation". I recommend watching that recording if you have not seen it

7. Summer 2019, major development goals ready and question authors started testing. Initial experiences were great and authors came up with unexpected patterns that led to new development.

8. During fall 2020, after long internal testing we will expose *Stateful* to students.

STACK & *Stateful* are tightly coupled but STACK will not do stateful matters. The authoring related issues with the "STACK with state" proof of concept made it pretty clear that adding state complicates things. No one wanted to break the balance of authoring simplicity and expressiveness of STACK by forcing state into it. Also, the authoring processes with state are much simpler if *Stateful* can just have its own question-model.

Stateful & STACK share most of the internal parts and (at least) one developer. As we share parts, we can now test new versions of various subsystems in either one before they turn to common features.
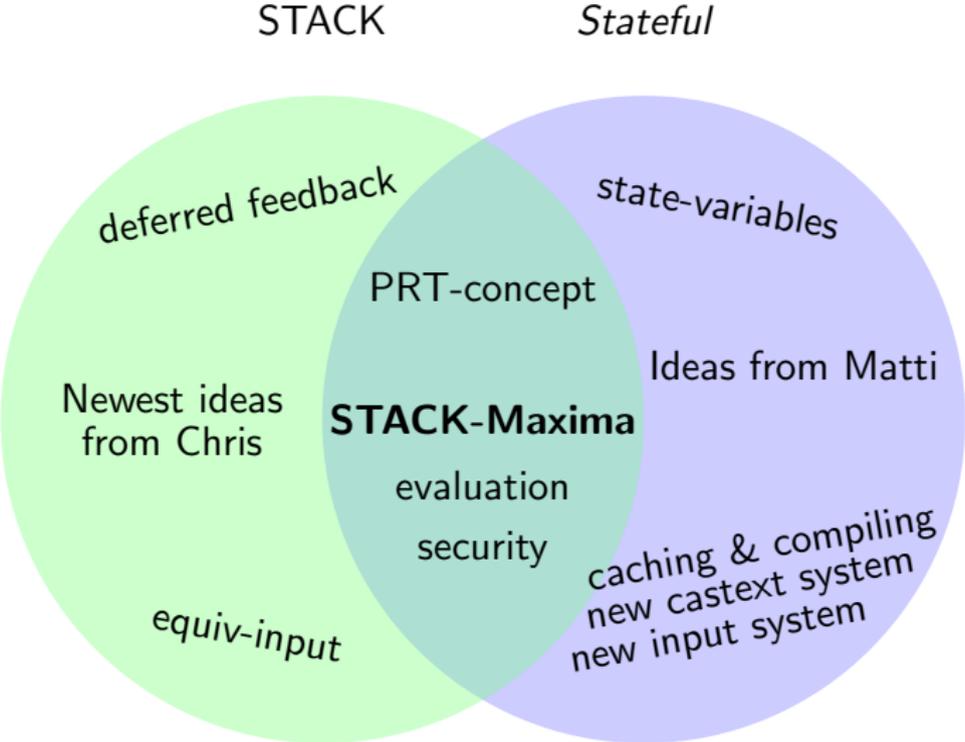
# Basic *Stateful* question terminology & logic 1/2

Stateful-questions are essentially just collections of parametric STACK questions that have been extended so that student input can lead to the question transitioning to another. When transitioning, one can change the values of state-variables, which are typically used as the parameters of those questions.

The common terms are:

scene   one of the questions in the collection

path   the sequence of scenes the student has visited as well as the transitions that led to those changes of scenes

# Basic *Stateful* question terminology & logic 2/2
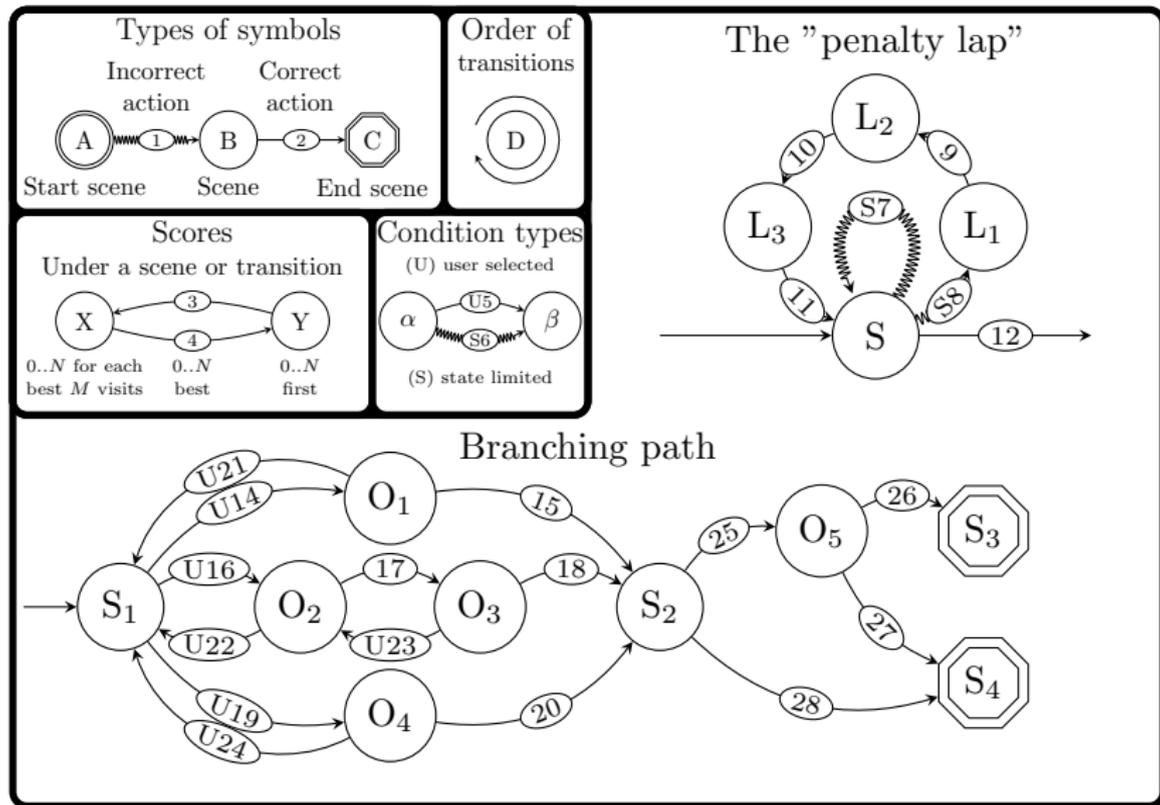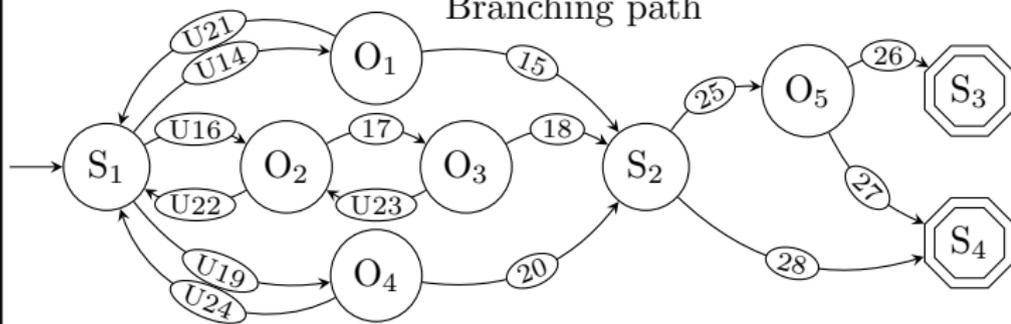
Some future plans

- External state, i.e., parameters coming from, e.g. learning analytics and means of storing parameters there.

- Visualisation and analysis tools for exploring the paths taken by large numbers of students. Where they drop out, which paths they choose, and on which paths do most people end up. Current research ideas aim for Sankey diagrams.
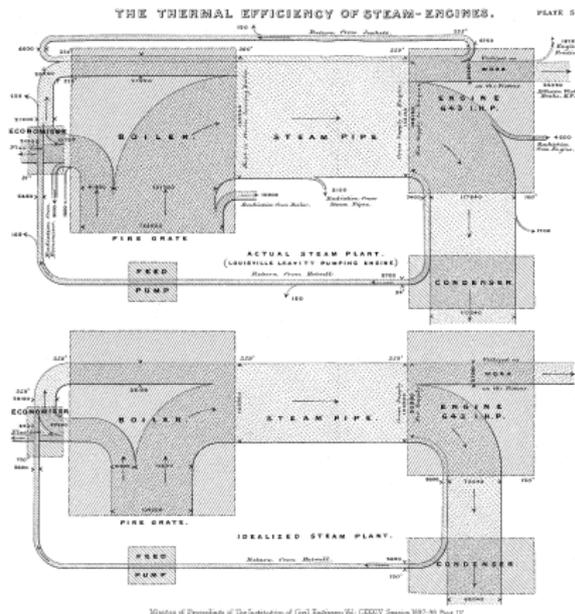
The original Sankey diagram from Wikipedia

This years sample question about integration 1/2

Back in 2016 the sample was an integration by parts question, which allowed the student to travel to the wrong direction infinitely. This time the question is simpler and has no infinite loops in the scene-graph. Now even simpler: a definite-integral. Now, what is the trap here?

Find out the area shaded in the plot below, between the polynomial $p(x) = x^3 - \frac{7 \cdot x^2}{6} - \frac{10 \cdot x}{3}$ and the x-axis when $x \in [-2, 3]$. Provide an exact value as the answer.

JSXGraph v1.1.0 Copyright (C) see https://jsxgraph.org
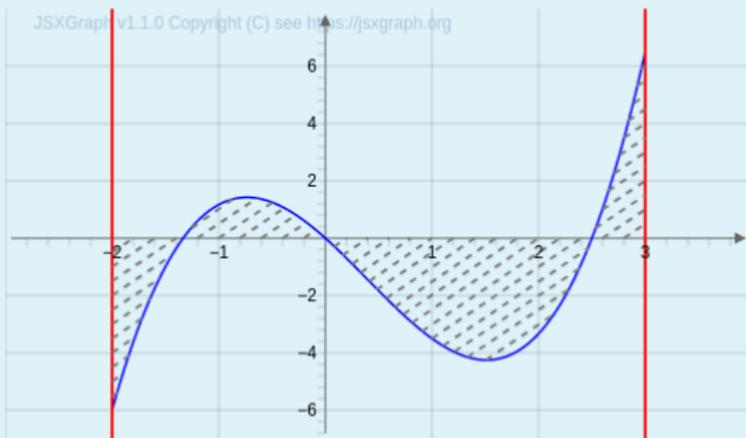
The area is: [   ]

This years sample question about integration 2/2
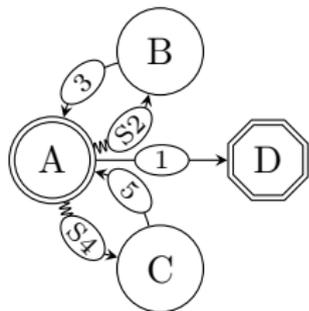
States

A) The question
B) Check indefinite integral of $p(x)$
C) Check sign changes, i.e., roots of $p(x)$
D) The end

State variables

`boolean thatError = false`
 if $\int_a^b p(x)\,\mathrm{d}x$ was given as an answer

`boolean otherError = false`
 if scene B was visited, unknown error

`boolean signError = false`
 if scene C was visited, any sign-error

Transitions

1) If correct answer, grant up to 1 point
 -0.1 for each error state visited
S2) If error & not visited & not sign-error
3) If correct indefinite integral
S4) If sign-error & not visited
5) If correct roots

The demo

The sample question is accessible though the EAMS site, after this presentation everyone attending the conference will also gain edit-rights so that they may try building *Stateful* questions. These rights will remain active for a short time during the conference.

The end

Stateful is available from GitHub but it does not come with the editor shown in the demo.
Should you be interested in *Stateful*, please contact me `matti.harjula@aalto.fi`. If the editor interests you, please contact my commercial alter-ego `matti.harjula@eleaga.fi`.

Thank you

Hopefully, a live demo can be arranged at the next international STACK conference, in Tallinn next spring.