

Tilallinen STACK $\Rightarrow$ *Stateful*

Matti Harjula

Aalto University

Älyoppi webinaari 6.11.2020

Ensimmäinen osa esityksestä

1. Pientä *Stateful* historiaa
2. *Stateful* ~~vs~~ & STACK
3. Pieni hengähdystauko, eli demo kysymys
4. Perus termistöä *Stateful* kysymyksiin

Statefullin lyhyt historia 1/2

1. 2012 STACK oli jo levinnyt laajalle Aallossa, mutta jotkut kaipasivat adaptiivisia tehtäviä, joita STACK ei yleisesti helpolla kyennyt tarjoamaan

Statefullin lyhyt historia 1/2

1. 2012 STACK oli jo levinnyt laajalle Aallossa, mutta jotkut kaipasivat adaptiivisia tehtäviä, joita STACK ei yleisesti helpolla kyennyt tarjoamaan
2. Adaptiivisuutta ja vaiheittaisuutta yritettiin rakentaa JavaScriptillä asiakkaan päässä, mikä oli työlästä ja helposti hajoavaa

Statefullin lyhyt historia 1/2

1. 2012 STACK oli jo levinnyt laajalle Aallossa, mutta jotkut kaipasivat adaptiivisia tehtäviä, joita STACK ei yleisesti helpolla kyennyt tarjoamaan
2. Adaptiivisuutta ja vaiheittaisuutta yritettiin rakentaa JavaScriptillä asiakkaan päässä, mikä oli työlästä ja helposti hajoavaa
3. 2015 muodostui ratkaisu jota sanotaan ”tilalliseksi STACKiksi”, se esiteltiin EAMS 2016 konferenssissa

Statefullin lyhyt historia 1/2

1. 2012 STACK oli jo levinnyt laajalle Aallossa, mutta jotkut kaipasivat adaptiivisia tehtäviä, joita STACK ei yleisesti helpolla kyennyt tarjoamaan
2. Adaptiivisuutta ja vaihteellisuutta yritettiin rakentaa JavaScriptillä asiakkaan päässä, mikä oli työlästä ja helposti hajoavaa
3. 2015 muodostui ratkaisu jota sanotaan ”tilalliseksi STACKiksi”, se esiteltiin EAMS 2016 konferenssissa
4. Mutta se oli epäkäytännöllinen ja tehtävien tuotanto ja testaaminen oli liian vaikeaa, positiiviset havainnot muutamasta testi kysymyksestä kuitenkin johtivat uusiin ideoihin...

Statefullin lyhyt historia 2/2

5. 2018 Älyoppihankkeesta löytyi rahaa toteuttaa kyseiset ideat, ja niitähän toteutettiin, *Stateful* on nyt uusi kysymystyyppi, joka riippuu STACKista ja toimii pitkälti samalla logiikalla

Statefullin lyhyt historia 2/2

5. 2018 Älyoppihankkeesta löytyi rahaa toteuttaa kyseiset ideat, ja niitähän toteutettiin, *Stateful* on nyt uusi kysymystyyppi, joka riippuu STACKista ja toimii pitkälti samalla logiikalla
6. Viimeisen parin vuoden aikana on päästy kokeilemaan tehtävän kehitystä ja se tuntuu nyt sujuvan paljon paremmin ja helpommin kuin 2015-16 mallissa

Statefullin lyhyt historia 2/2

5. 2018 Älyoppihankkeesta löytyi rahaa toteuttaa kyseiset ideat, ja niitähän toteutettiin, *Stateful* on nyt uusi kysymystyyppi, joka riippuu STACKista ja toimii pitkälti samalla logiikalla
6. Viimeisen parin vuoden aikana on päästy kokeilemaan tehtävän kehitystä ja se tuntuu nyt sujuvan paljon paremmin ja helpommin kuin 2015-16 mallissa
7. Tänä syksynä ensimmäinen tehtävämateriaali päättyi opiskelijoiden eteen, Kimmo Kulmala kertoo siitä enemmän tämän esityksen toisessa osassa

STACK ja Stateful

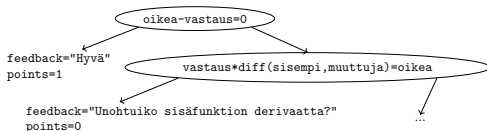
Muuttujat

```
muuttuja: rand([x, y, u, v, t]);  
sisempi: rand([sin(muuttuja), cos(muuttuja)]);  
ulompi: rand([sqrt(sisempi), sisempi^3]);  
lauseke: ulompi;  
oikea: diff(lauseke, muuttuja);
```

Kysymysteksti

```
Derivoippa {@lauseke@} {@muuttuja@}:n suhteen:  
  
\(\frac{\text{term}{d}}{\text{term}{d}}{\text{term}{d}}{@muuttuja@}...\ = \\  
[[input:vastaus]]  
[[validation:vastaus]]  
  
[[feedback:tarkastus]]
```

Tarkastuspuu



- ▶ STACK tehtävät ovat usein parametrisia
- ▶ Nämä parametrit ovat satunnaisia, eikä niitä voi tehtävän aikana muuttaa

STACK ja Stateful

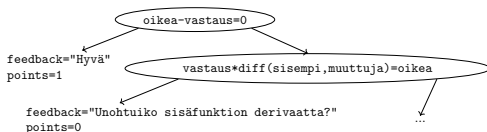
Muuttujat

```
muuttuja: rand([x, y, u, v, t]);  
sisempi: rand([sin(muuttuja), cos(muuttuja)]);  
ulompi: rand([sqrt(sisempi), sisempi^3]);  
lauseke: ulompi;  
oikea: diff(lauseke, muuttuja);
```

Kysymysteksti

```
Derivoippa {@lauseke@} {@muuttuja@}:n suhteen:  
  
\(\frac{\text{term}{d}}{\text{term}{d}}\{{@muuttuja@}\}\dots = \backslash  
[[input:vastaus]]  
[[validation:vastaus]]  
  
[[feedback:tarkastus]]
```

Tarkastuspuu



- ▶ STACK tehtävät ovat usein parametrisia
- ▶ Nämä parametrit ovat satunnaisia, eikä niitä voi tehtävän aikana muuttaa
- ▶ *Stateful* tehtävissä niitä voi muuttaa tarkastuspuussa

STACK ja Stateful

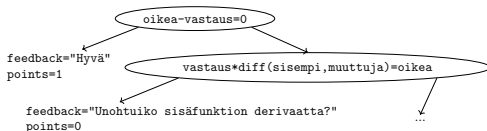
Muuttujat

```
muuttuja: rand([x, y, u, v, t]);  
sisempi: rand([sin(muuttuja), cos(muuttuja)]);  
ulompi: rand([sqrt(sisempi), sisempi^3]);  
lauseke: ulompi;  
oikea: diff(lauseke, muuttuja);
```

Kysymysteksti

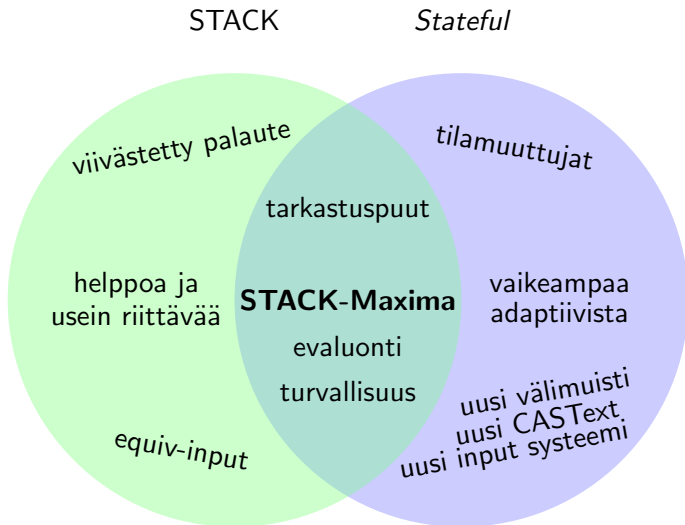
```
Derivoippa {@lauseke@} {@muuttuja@}:n suhteen:  
  
\(\frac{\text{d}}{\text{d}}\{@muuttuja@\}\dots = \\  
[[input:vastaus]]  
[[validation:vastaus]]  
  
[[feedback:tarkastus]]
```

Tarkastuspuu



- ▶ STACK tehtävät ovat usein parametrisia
- ▶ Nämä parametrit ovat satunnaisia, eikä niitä voi tehtävän aikana muuttaa
- ▶ *Stateful* tehtävissä niitä voi muuttaa tarkastuspuussa
- ▶ *Stateful* tehtävissä voi lisäksi olla useampi STACK tyylinen kysymys ja muutoksen yhteydessä voidaan vaihtaa esitettävää kysymystä

Stateful ~~vs.~~ & STACK



Perus *Stateful* terminologiaa 1/2

Stateful tehtävät ovat käytännössä kokoelmia parametrisista STACK tehtävistä, joiden tarkastuspuihin on lisätty tilasiirtymiin liittyvää logiikkaa. Kun opiskelijan syöte aktivoi tarkastuspuun tietyn haaran voi tapahtua tilasiirtymä, jonka yhteydessä tehtävän tilamuuttujien arvoja voidaan päivittää ja mahdollisesti esitettävää tehtävää vaihtaa. Tilamuuttujat vastaavat perinteisen STACK tehtävän parametreja.

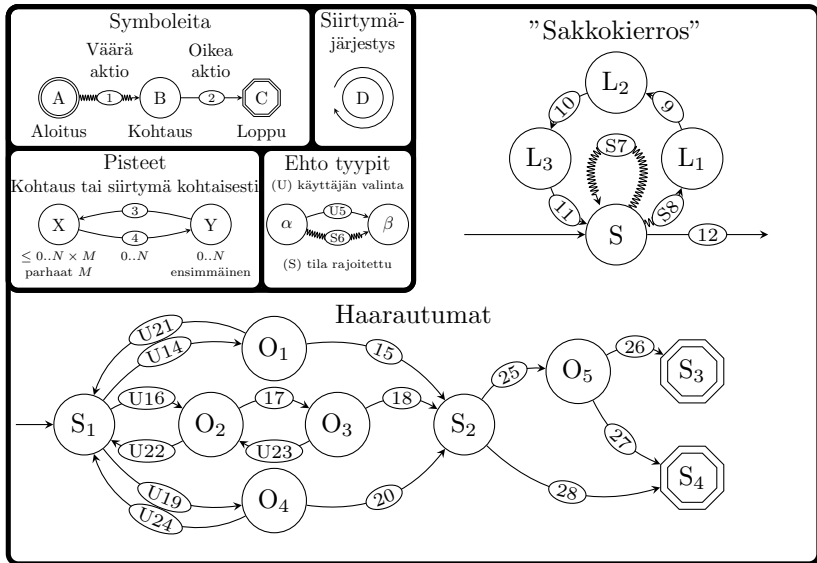
Käytössä olevia termejä:

kohtaus on joku kokoelman tehtävistä tietyssä tilassa

polku on sarja kohtauksia, joiden läpi opiskelija on kulkenut sekä lista niistä siirtymistä, jotka aktivoituivat tällä matkalla

tila tilamuuttujien arvojen kombinaatio, mukaanlukien polku ja nyt esillä oleva tehtävä

Perus *Stateful* terminologiaa/logiikkaa 2/2



Tulevaisuutta

Tämä ei ole lautapeli... Nyt on Kimmon vuoro esitellä tehtävä jonka suorituksista tämä on piiretty.

