**Smartschool tracking**

Ik ben hier uitgegaan van de bestaande tracking van Xerte, omdat wij dat op zeer korte termijn kunnen implementeren.

De tracking van Xerte wordt bijgehouden in javascript structuren. Deze state kan in zijn geheel als JSON script gestuurd worden en ook kan de state in 1 klap gezet worden op basis van een JSON string

Elke 'pagina' in Xerte kan 0 of meerdere 'interacties' bevatten

De state van een 'pagina' wordt opgeslagen in de volgende structuur:

function InteractionTracking(page\_nr, ia\_nr, ia\_type, ia\_name)

{

this.page\_nr = page\_nr; // Page id

this.page\_ref = page\_nr+1; // Pagina nr

this.ia\_nr = ia\_nr; // Interactiviteit id (-1 voor de pagina zelf)

this.ia\_ref = ia\_nr+1;

this.ia\_type = ia\_type; // Soort interactiviteit: 'match', 'multiplechoice', 'numeric', 'fill-in', 'text' en 'page'

this.ia\_name = ia\_name; // Naam (referentie naar de interactiviteit)

this.state = "entered"; // Status: 'entered', 'exited'

this.start = new Date(); // Start

this.end = this.start; // Einde (als state == 'exited')

this.count = 0; // Hoe vaak bezocht

this.duration = 0; // Tijdsduur (als state == 'exited')

this.nrinteractions = 0; // Aantal interactions (als ia\_nr == -1, m.a.w. dit is de pagina)

this.weighting = 0.0; // Gewicht (voor gewogen gemiddelde)

this.score = 0.0; // Score (0.0 - 100.0)

this.result = 'unknown'; // Resultaat: 'unknown', true, false voor interactiviteit match, multiplechoice en fill-in

this.complete = false; // Alle interactiviteiten gedaan?

this.correctoptions = ""; // Correcte opties (in de vorm van a,b,c) voor match, multiplechoice en fill-in

this.correctanswer = ""; // Correcte antwoorden voor match, multiplechoice en fill-in

this.correctfeedback = ""; // Feedback van correcte antwoord

this.learneroptions = ""; // Gegeven antwoordopties (in de vorm van a,b,c) voor match, multiplechoice en fill-in

this.learneranswer = ""; // Gegeven antwoorden

this.answerfeedback = ""; // Feedback over de gegeven antwoorden

this.id = ""; // uid

}

Welke velden zijn ingevuld hangt af van het type interactiviteit:

1. **Pagina (dus ia\_nr => -1)**

Twee gevallen:

* 1. Geen interactiviteiten op de pagina: Score is 100 als de pagina langer dan 1 seconde bezocht is
  2. Wel interactiviteiten op de pagina: Gemiddelde score van de interactiviteiten.

1. **match**

Wordt gebruikt bij drag en drop oefeningen.

correctoptions en learneroptions is gevuld met een array van objecten { source, target }

1. **multiplechoice**

Wordt gebruikt door mcq en quiz. Bij mcq is er een vraag per pagina, bij een quiz zijn er meerdere vragen op 1 pagina

correctoptions en learneroptions zijn gevuld met indicatie van antwoorden (1, 2, 3)

correctanswer en learneranswer zijn gevuld met een text representatie van de antwoorden

correctfeedback en learner feedback zijn ook gevuld

1. **fill-in**

Wordt o.a. gebruikt bij fill in the gaps

correctoptions en learneroptions zijn niet gevuld met een array van objecten { source, target }

correctanswer en learneranswer zijn gevuld met een text <source> --> <target>

correctfeedback is ook gevuld

1. **text**

wordt nog niet gebruikt, maar is bedoeld voor open vragen

De complete state wordt opgeslagen in de volgende structuur:

function TrackingState()

{

this.initialised = false; // Geïnitialiseerd?

this.mode = ""; // Normal is normale tracking

this.currentid = ""; // Huidige interactie id -> wordt gebruikt bij restore state

this.currentpageid = ""; // Huidige pagina id -> wordt gebruikt bij restore state

this.trackingmode = "full"; // Tracking mode: 'none', 'minimal', 'full'

this.scoremode = 'first'; // Score mode: 'first', 'last'

this.nrpages = 0; // Aantal pagina's

this.pages\_visited=0; // Aantal pagina's bezocht. Wordt gebruikt voor completeness, maar gaat veranderen

this.start = new Date(); // Start tijd

this.duration\_previous\_attempts = 0; // Totale tijd in vorige pogingen

this.lo\_type = "pages only"; // Leer object type: 'pages only', 'interactive'

this.lo\_passed = -1.0; // Score van 0.0 - 100.0 wanneer een object geslaagd is (standaard 55)

this.lo\_completed = "unknown"; // Compleet: 'unknown', 'incomplete' en 'completed'

this.finished = false; // Gestopt

this.interactions = new Array(); // Array van InteractionTracking

}

Ik stel het volgende voor

1. Er komt een speciaal entry point voor Smartschool, bijvoorbeeld https://<url van installatie>/smartschool.php?template\_id=32. In deze call wordt er een OAuth gedaan met smartschool, om te zien welke student cq. docent is ingelogd, en de user id wordt meegegeven aan de Xerte runtime.
2. Bij het starten van het leerobject wordt er gevraagd (via webservice call) naar state (als het een resume betreft). Smartschool moet dat op een of andere manier aangeven. Het leerobject gaat dan verder waar ze gebleven waren. Hier wordt de hele JSON string meegeven die de vorige keer werd meegegeven bij het afbreken. (zie onder)
3. Afhankelijk van wat jullie willen, kan er een webservice call bij de start van een pagina maar in ieder geval bij het verlaten van de pagina gestuurd worden met een InteractionTracking object JSON
4. Bij het afsluiten een call met de status (complete, suspended) met de score of de suspend state met een TrackingState object in JSON

Ik heb nog geen webservice interface bedacht, maar REST zou verre weg het makkelijkst zijn, want die kunnen we rechtstreeks vanuit javascript gebruiken.

Ook zullen we vast hier en daar wat details moeten toevoegen over hoe we een uid aan een leerobject hangen. En zaken als publiceren en dergelijke.