



Systém automatické kontroly odevzdávaných semestrálních prací (SAKo)

Petr Neduchal¹, Lukáš Bureš²

1 Úvod

Jak studentům i sobě usnadnit odevzdání semestrální práce tak, aby nemuseli řešit způsob odeslání? Nebylo by také hezké zprostředkovat jim automaticky alespoň orientační hodnocení bez toho, aby museli čekat na opravu práce vyučujícím? A jak to všechno ideálně udělat v jednom jediném kroku?

Přesně to jsou otázky, které jsme si s kolegou začali pokládat při výuce předmětů. Postupem času se objevila myšlenka vytvoření systému pro automatickou kontrolu odevzdávaných prací (dnes pojmenovanou zkráceně SAKo). Impulzem k vytvoření byla ještě další otázka. *Jsmo vůbec schopni takový systém vytvořit?* Jak se ukázalo tak jsme.

2 O systému SAKo

SAKo je systém, který je zaměřen na semestrální práce naprogramované v jazyce Python. Odevzdávanou úlohu je schopný jednak nahrát na server a navíc je schopný spustit validaci, která studentovi přidělí body a tím mu dá představu o tom, jestli se výsledky semestrální práce shodují s výsledky, které systém očekává.

Přístup, který jsme zvolili má i své nevýhody. Například semestrální práce musí mít jasně zadaný výstup a pokud by bylo cílem použít výhradně automatickou validaci bez následné opravy vyučujícím, pak musí existovat přesná ground truth data, jejichž dosažení by znamenalo získání plného počtu bodů.

Systém SAKo je založen na protokolu HTTP, tedy na stejném jako komunikace při prohlížení webových stránek na internetu. Celý systém je rozdělen na serverovou část a klienta, obě části budou popsány v následujících odstavcích.

3 Serverová část

Server systému SAKo představuje spolupráci dvou skriptovacích jazyků. Po navázání spojení se nejdříve zavolá skript v jazyce PHP (viz Vrána (2010)), který se stará o bezpečnou (viz Huseby (2006)) autentizaci odevzdávajícího uživatele, kontrolu existence dané úlohy a splnění jejich parametrů. V dalším kroku je zavolán python skript Bandit¹, který slouží ke kontrole bezpečnosti nahraných skriptů. Navíc je ho možné nastavit v přehledném konfiguračním souboru.

Poté je řízení předáno validátoru, který má každá úloha vytvořený na míru. Tato část

¹ studentka navazujícího (bakalářského, doktorského) studijního programu Aplikované vědy a informatika, obor Kybernetika, specializace Počítačové vidění, e-mail: neduchal@ntis.zcu.cz

² student doktorského studijního programu Aplikované vědy a informatika, obor Kybernetika, specializace Počítačové vidění, e-mail: lbures@ntis.zcu.cz

¹<https://github.com/stackforge/bandit>

je implementována v programovacím jazyce Python. Úkolem této části je spustit kód nahraný uživatelem a zhodnotit jeho výstup. S tím souvisí další obtíž při zadávání úlohy. Je totiž nutné, aby soubory a obsažené funkce (resp. třídy a metody) měly přesně zadané neměnné názvy, parametry a výstupy. Vzhledem k opravování úlohy je taková definice na druhou stranu spíše výhodou.

3.1 Definice validátoru

K definování validátoru slouží v systému SAKo v současné době dva základní soubory. Skript `validator.py`, který obstarává samotnou validaci a využívá služeb knihovny `SAKoTools` - tedy knihovny starající se o načítání uživatelských skriptů ve správném adresáři atp. Druhým souborem je `validator.json`, ve kterém je uvedeny informace o validátoru a úloze. V současné době má soubor následující tvar:

```
{
  "name": "Copy",
  "desc": "Kopírování souboru na server",
  "open": "2015-03-01 00:00:00",
  "deadline": "2015-12-02 00:00:00",
  "bantime": "2"
}
```

Popořadě je zadáno jméno validátoru, jeho popis, datum a čas otevření a uzavření validace úlohy a čas po který není možné opakovat odevzdání. Tím je vytvořen validátor a nyní je možné využít klienta k odevzdání úlohy.

4 Klientský skript

Klient systému SAKo se ve verzi 1.0. sestává z jedné třídy obstarávající práci se složkou a z odevzdávací funkce `submit`, jejíž volání po připojení klienta SAKo vypadá následovně:

```
SAKo.submit('zarazeni', 'adresar', 'login', 'heslo', 'uloha')
```

Parametry mají zleva následující význam. Zařazení studenta do předmětu (např. *zdo*), cesta k adresáři se soubory k odevzdání, přihlašovací jméno, heslo a nakonec název úlohy, kterou chce student odevzdat. Název úlohy musí být definovaný v zadání úlohy.

5 Závěr

System je již v testovacím provozu nasazen. V současné době je připravovaná nová verze serveru i klienta. Na straně klienta bude hlavní novinkou možnost volat přímo skript `SAKo.py` a ne jen metodu `submit`. Na straně serveru pak dojde k rozšíření definice validátoru, který bude napojen na systém upozorňující na odevzdané práce a dále je v plánu vytvoření subsystému pro automatický výpočet a vizualizaci výsledků a statistik ve webovém rozhraní systému.

Literatura

- Vrána, J., 2010. *1001 tipů a triků pro PHP*. Cpress, Brno, ISBN 978-80-251-2940-1.
Huseby, S. H., 2006. *Zranitelný kód*. Cpress, Brno, ISBN 80-251-1180-6.