Titelfolie

* Traditionelle Multiple-Choice-Quizzes sieht man heute überall und in allen Lebenslagen
* Egal ob A-Teil Matheklausur, bei Wer wird Millionär oder bei den Umfragen in der Straßenbahn, die sich eh keiner durchliest – wir kennen sie alle
* Legenden zufolge mussten schon damals im alten Ägypten die Bestatter schon Tests durchführen, um die Mumien auseinanderzuhalten   
  **FOLIE WEITER**(Es ist übrigens B)
* aber darum geht es ja gar nicht
* alle diese Beispiele haben ein großes Problem: man muss sich die Fragen mit viel Mühe selbst ausdenken und genau damit habe ich mich auseinandergesetzt, indem ich mich mit der **FOLIE WEITER**
* Automatische Erstellung von Quizfragen aus einer Ontologie von Krankenhausinformationssystemen
* beschäftigt habe.

Gliederung

* Grundidee – um was geht es eigentlich?
* Grundlagen & SNIK
* Umsetzung des Programms
* Demonstration der Funktionsweise
* Ergebnis
* Ausblick  
  **FOLIE WEITER**

Grundidee

* Entwicklung eines Multiple-Choice-Quizzes, welches automatisch seine eigenen Fragen generiert
* Modell der Krankenhausinformationssysteme, um eine einfache Lern- und Abfrageplattform für Medizinstudenten zu entwickeln
* Daten der Quizgenerierung sollten hierbei aus der bestehenden SNIK-Ontologie entnommen werden
* Ausgabe der Quizfragen soll in einer nutzerfreundlichen Form geschehen
* Doch was ist eigentlich SNIK? Und was ist denn eine Ontologie?

**FOLIE WEITER**

Grundlagen 1 – Was ist eine Ontologie?

* im informatischen Kontext: eine sprachlich gefasste und geordnete Darstellung einer Menge von Begrifflichkeiten mit festen Beziehungen untereinander
* Beziehungen unter den Begrifflichkeiten sind hierbei formuliert in **RDF-Tripeln**
* **Was bedeutet das?**
* **RDF-Tripel** bilden eine Aussage, die aus Subjekt, Prädikat und Objekt zusammengesetzt ist
* Subjekt bildet hierbei das Grundelement der Beziehung
* Objekt beschreibt das Element, auf das sich bezogen wird
* Prädikat beschreibt schlussendlich den Bezug zwischen Subjekt und Objekt
* hier am Beispiel: New York liegt in USA ERLÄUTERN (NICHT umkehrbar!!)
* hat man nun eine Menge dieser Tripel, so lässt sich aus den Beziehungen ein Netz aufspannen: **FOLIE WEITER**

Grundlagen 2 – Was ist eine Ontologie?

* auf den ersten Blick sind hier verschiedene Elemente sichtbar

((Orange Konzepte, Blau Spezifikationen, hellblau explizite –``-))

* -> (Aufzählen)
* dieses Modell ähnelt hier einem Netzwerkmodell aus der Informatik
* Anders als bei diesem Modell besitzen diese Elemente jedoch hier Beziehungen untereinander, die mit einem Pfeil aufgezeigt werden
* viele dieser Beziehungen sind in RDF standartisiert, es muss jedoch nicht immer nach Standart gehen
* -> (Beispiele nennen)
* rdfs:subClassOf -> beispielsweise nichts anderes als „Unterklasse von“
* a -> „ist ein“
* (Subjekt Prädikat Objekt Beispiel) RDF-Modell
* auch die SNIK-Ontologie ist so aufgebaut: **FOLIE WEITER**

Grundlagen 3 – SNIK

* was ist eigentlich SNIK?
* SNIK bedeutet „Semantisches Netz des Informationsmanagements im Krankenhaus“
* ist eine Ontologie mit der Zielsetzung, Begriffe des Informationsmanagements im Krankenhaus und deren Beziehungen untereinander zu beschreiben
* sieht selbst die Nutzung als Hilfsmittel für Medizinstudenten vor
* SNIK-Ontologie wurde am IMISE (Institut f. medizinische Informatik) entwickelt
* SNIK Ontologie ist öffentlich zugänglich und Open Source, sodass jeder Interessent daran teilhaben kann -> bietet öffentliche Endpunkte für Drittanwendungen an (((Application Programming Interface))) **FOLIE WEITER**

Umsetzung des Programmes

* für meinen Nutzungszweck habe ich hier den SPARQL-Endpunkt für die Rohdaten meiner Fragengenerierung genutzt
* kurz: SPARQL ist eine SQL ähnliche Sprache, die mit standartisierten RDF-Beziehungen arbeitet
* hier lassen sich nun verschiedene Abläufe entwickeln
* am Beispiel der DEFINITION-Query hier vorgestellt:
* als erstes muss eine Strategie erstellt und geplant werden, sozusagen was die Query machen soll
* bei der DEFINITION-Query wird ein Objekt gesucht, welches eine Definition aufweist ( Was wird durch X definiert? )
* Tripel sieht hier folgendermaßen aus -> Definition definiert Objekt
* -> Fragenkopf fertig
* nun müssen falsche Antworten generiert werden
* dazu werden einfach drei Objekte aus dem näheren Umfeld genutzt
* -> Distraktoren sind so ebenfalls erledigt
* nun muss das Ganze in eine Query eingefügt werden, die ich jetzt nicht näher erläutern werde
* hier kommen zu der grundlegenden Strategie noch verschiedene Filter hinzu, welche zum Beispiel Ausgabesprache oder Textlänge filtern
* Query kann nun im SPARQL-Endpunkt ausgeführt werden
* Endpunkt durchläuft die Query und gibt am Ende eine Ergebnismenge von zu der Query passenden Tripeln in Rohdatenform aus.
* Das sieht wie folgt aus:
* corrAns (correct answer) ist das Objekt, zu dem die Definition gegeben ist  
  -> „Sample“
* cDef (correct definition) ist das Subjekt, welches dieses Objekt definiert  
  -> „What is defined by „Is a specimen taken from a patient […]” ?
* d1-3 (distractor 1-3) sind Nebenobjekte, die ablenken sollen
* nun muss im letzten Schritt die Ergebnismenge nur noch formatiert und verarbeitet werden, um die Fragen in korrekter Satzform anzuzeigen
* die Ergebnismenge wird hier also durch ein Python-Programm weiterverarbeitet und zur finalen Frage umgeschrieben
* die Fragen werden in einem Array zwischengespeichert und am Ende, wie hier klein zu sehen, dem Nutzer in der Kommandozeile oder wahlweise nach Ausgeben im Programm QuizMaster angezeigt
* was der Nutzer davon mitbekommt, werde ich jetzt demonstrieren: **FOLIE WEITER**

Demonstration / „Vorführung“

* der Nutzer wird nach Öffnen des Programmes um Auswahl des Fragetyps gebeten
* nach Auswahl des Querytyps, hier also Definition, wird nun zusätzlich noch die Ausgabe in Konsole oder als Datei abgefragt
* ist der Ausgabetyp bestätigt, sendet das Programm eine http-GET-Anfrage an den Endpunkt mit der gewünschten Query, einem Zeitlimit, nach dem die Abfrage abbricht und weiteren Daten und wartet das Zeitlimit ab, bis es ein Ergebnis erhält
* das Ergebnis wird am Ende angezeigt und das Programm kann geschlossen werden **FOLIE WEITER**

Ergebnis

* Entstanden ist also am Ende ein plattformunabhängiges Programm, welches es einem Nutzer ermöglicht, ohne jegliche Kenntnisse über Ontologien, SNIK, RDF oder Python randomisiert Fragen generieren zu lassen
* durch das Paket py2exe ist in einer Windows-Umgebung nicht einmal die Installation von Python selbst nötig und eine simple .exe – Datei genügt
* durch die Möglichkeit des Exports der Fragen in eine Datei ist es möglich, diese in Quizprogrammen wie dem von mir in der 10. Klasse entwickelten QuizMaster zu öffnen und diese so mit einer GUI zu beantworten
* Abschließend kann man also sagen, dass sich die Bestatter im alten Ägypten bei ihrer Aufnahmeprüfung durch eine Ontologie eine Menge Arbeit hätten sparen können. **FOLIE WEITER**

(kleiner) Ausblick

* mögliche API für Quizprogramme, um die Kommandozeilenumgebung im Hintergrund laufen zu lassen
* Möglichkeit der Übersetzung der Fragen und Antworten in andere Sprachen
* einige Problembehebungen -> die Klasse „entity type“ taucht bei vielen Fragen als Antwort auf, da sie erst kürzlich zur Ontologie hinzugefügt wurde
* mehr Algorithmen zur Fragengenerierung
* nun sage ich Vielen Dank fürs Zuhören! Gibt es noch Fragen?