

Kurzanleitung zur Benutzung des SceneMaker2VOnDA-Compilers

Simon Ahrendt, Max Depenbrock, Jana Jungbluth

20. September 2018

1 Überblick

Dies ist eine kurze Beschreibung, wie man den SceneMaker2VOnDA-Compiler benutzt, um einen mittels *Visual SceneMaker* erstellten Dialog in das VOnDA-Framework zu übersetzen.

Dabei wird der SceneMaker-Dialog durch einen sogenannten *Sceneflow* definiert (wird von Visual SceneMaker automatisch als `sceneflow.xml` im Projekttordner generiert). Der Output unseres Compilers hingegen besteht aus `.rudi`-Dateien, die VOnDA-Code enthalten, und aus einer Datei `ontology.nt`, die alle notwendigen Datenbankeinträge enthält.

Um den Compiler zu verwenden, sollte das Shell-Skript `start_script.sh` im Ordner `Compiler` ausgeführt werden. Dieses Skript erhält als erstes Argument den Pfad zum *Sceneflow*, der übersetzt werden soll, und als zweites Argument den Pfad zu einem (bereits existierenden) Ordner, in dem der Output abgelegt werden soll.

Wichtig ist die Unterscheidung zwischen der Generierung in einem vorhandenen VOnDA-Projekt bzw. ohne ein bereits existierendes VOnDA-Projekt. Durch setzen der Flag `-b` kann angezeigt werden, dass der angegebene Output-Ordner bereits ein rudimentär funktionsfähiges VOnDA-Projekt enthält, dem nur noch die `.rudi`-Dateien und die Ontologiedatei fehlen, um kompilierfähig zu sein. In diesem Fall werden die generierten Dateien vom SceneMaker2VOnDA-Compiler an die richtigen Stellen gesetzt und das VOnDa-Projekt anschließend kompiliert.

2 Generierung in beliebigem Output-Verzeichnis

Falls die vom SceneMaker2VOnDA-Compiler generierten Dateien nicht in ein schon existierendes VOnDA-Projekt eingebaut, sondern einfach im angegebenen Output-Ordner abgelegt werden sollen, so sollte `start.sh` ohne die Flag `-b` ausgeführt werden.

In diesem Fall erstellt der Compiler im Output-Ordner die Unterordner `Rudi`, in dem alle generierten `.rudi`-Dateien (eine für jeden im Sceneflow definierten Superknoten) abgelegt werden, und `Ontology`, in dem die Datei `ontology.nt` abgelegt wird.

3 Generierung in vorhandenem VOnDA-Projekt

Falls `start.sh` mit der Flag `-b` ausgeführt wird, sollte der angegebene Output-Ordner bereits die Struktur eines VOnDA-Projekts besitzen und sich in einem Unterordner des VOnDA-Hauptverzeichnisses befinden (beispielsweise in `vonda/examples`).

Für das Gerüst eines VOnDA-Projekts kann man beispielsweise den Ordner `VOnDA_ExampleProject` als Grundlage verwenden. In diesem müssen nur noch folgende Änderungen vorgenommen werden, bevor das `start.sh`-Skript ausgeführt werden kann:

- Implementierung von NLG und NLU, die mit der Verwendung im *Sceneflow* übereinstimmen
- Unter `VOnDA_ExampleProject/src/main/java/de/dfki/introduction/` die Datei `ChatAgent.java` bearbeiten:
 - Jeden im *Sceneflow* definierten Superknoten (inklusive `mainAgent`) als Attribut der Klasse `Rdf` definieren (z.B. `Rdf mainAgent;`)
 - Jeden dieser Superknoten in der `init`-Methode dem zugehörigen Datenbankobjekt zuordnen (z.B. `mainAgent = proxy.getRdf("<cat:mainAgent>");`)

Wer `VOnDA_ExampleProject` nicht als Grundlage verwenden möchte, sollte selbst ein funktionsfähiges VOnDA-Projekt erstellen können. In diesem Fall ist nur wichtig zu wissen, dass die Ausführung des `SceneMaker2VOnDA`-Compilers mit der Option `-b` voraussetzt, dass das Projekt die Unterordner `src/main/rudi/` und `src/main/resources/ontology` enthält, und letzterer eine `.ini`-Datei beinhaltet, die einen Default-Namespace definiert.

Waren das Kompilieren und Bauen erfolgreich, kann der Dialog durch das Shell-Skript `openGui` gestartet werden.

4 Post-Processing und die `-n` Flag

Da `Visual SceneMaker` keine Rauten im Code-Segment erlaubt, diese aber zur Bezeichnung von Dialogakten in VOnDA benötigt werden, versucht der `SceneMaker2VOnDA`-Compiler an geeigneter Stelle die Raute zu ergänzen, so dass bei der Dialogerstellung im `Visual SceneMaker` diese einfach weggelassen werden kann.

In Fällen, wo dieses Verhalten nicht erwünscht ist (insbesondere falls zu einem zukünftigen Zeitpunkt `Visual SceneMaker` durch einen anderen graphischen Editor ersetzt werden sollte), kann dieses Post-Processing durch Setzen der Flag `-n` ausgeschaltet werden.

5 Beispielprojekt *Introduction*

Der Ordner `scenemaker_projects` enthält den Beispieldialog *Introduction*. Dieser kann verwendet werden, um den `SceneMaker2VOnDA`-Compiler zu testen.

Nach Befolgung der Schritte unter Punkt 3) ist es dabei aber noch notwendig, manuell zwei Zeilen in der erstellten Datei `Hello_node.rudi` zu verändern, da der korrekte Code von *Visual SceneMaker* leider nicht richtig verarbeitet werden kann. Beide Änderungen betreffen die Regel `wait_node`:

- Der Vergleich `lastDA() == "positive"`
muss zu `lastDA().what == "positive"` abgeändert werden
- Der Vergleich `lastDA() == "negative"`
muss zu `lastDA().what == "negative"` abgeändert werden