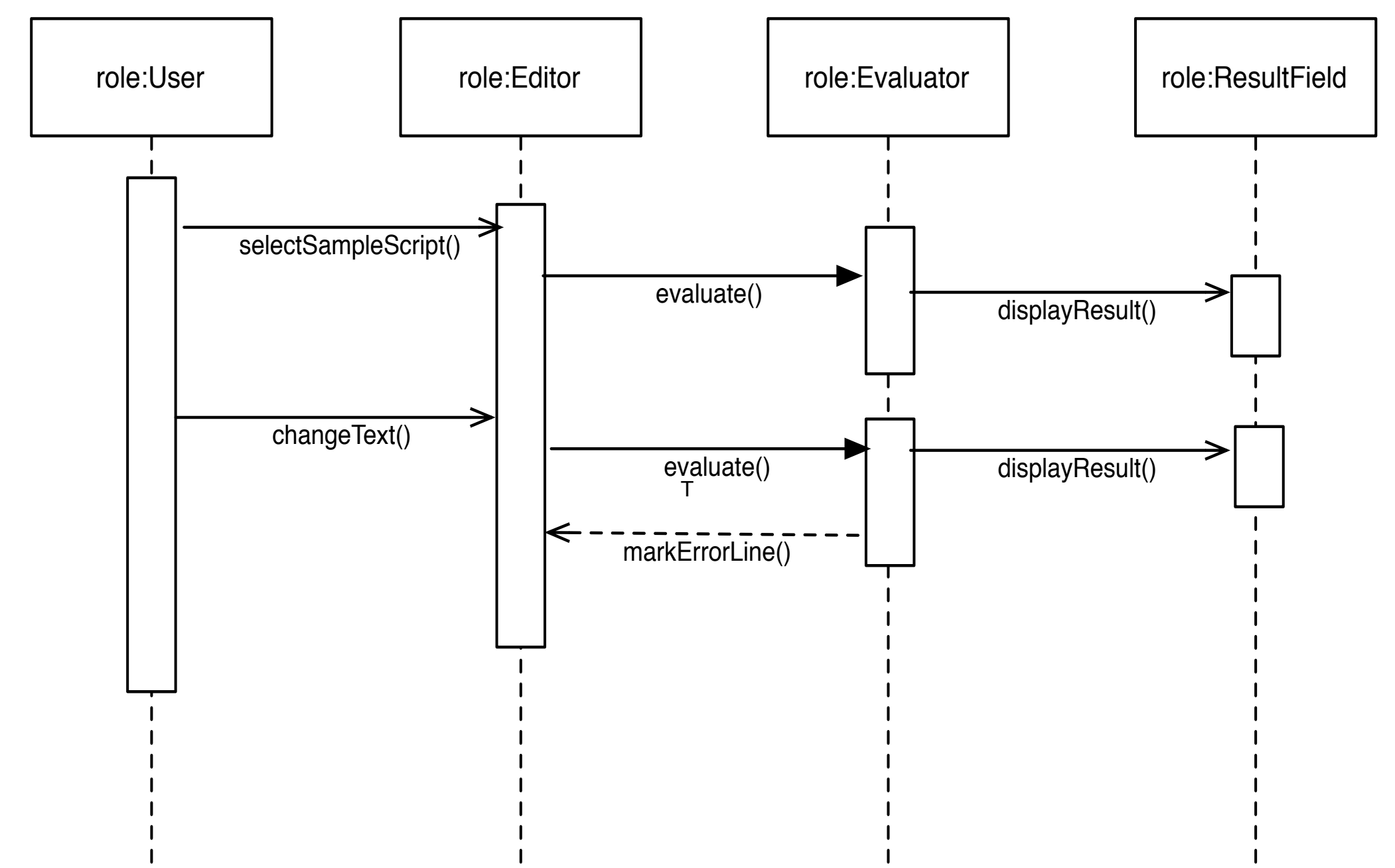


Anforderungen

- einbettbarer Texteditor
- verschiedene Codeauszeichnungen
- Bereitstellung von Beispielscripten
- Evaluierung des Codes
- Ausgabe des ausgeführten Codes
- Markierung einer fehlerhaften Zeile

Anwendungsfall



Dieses Sequenzdiagramm zeigt einen typischen Anwendungsfall für die Interaktion mit dem Editor sowie der Evaluierung des eingegeben Codes.

Umsetzung der Codeauszeichnung

Tcl - Commands	✓
Textbereiche	✓
Kommentare	✓
Zeilenumbrüche	✓
Sonderzeichen	✓
Auszeichnung besonderer Wörter	✓
Codeauszeichnung in Textbereichen	X

Graphische Umsetzung

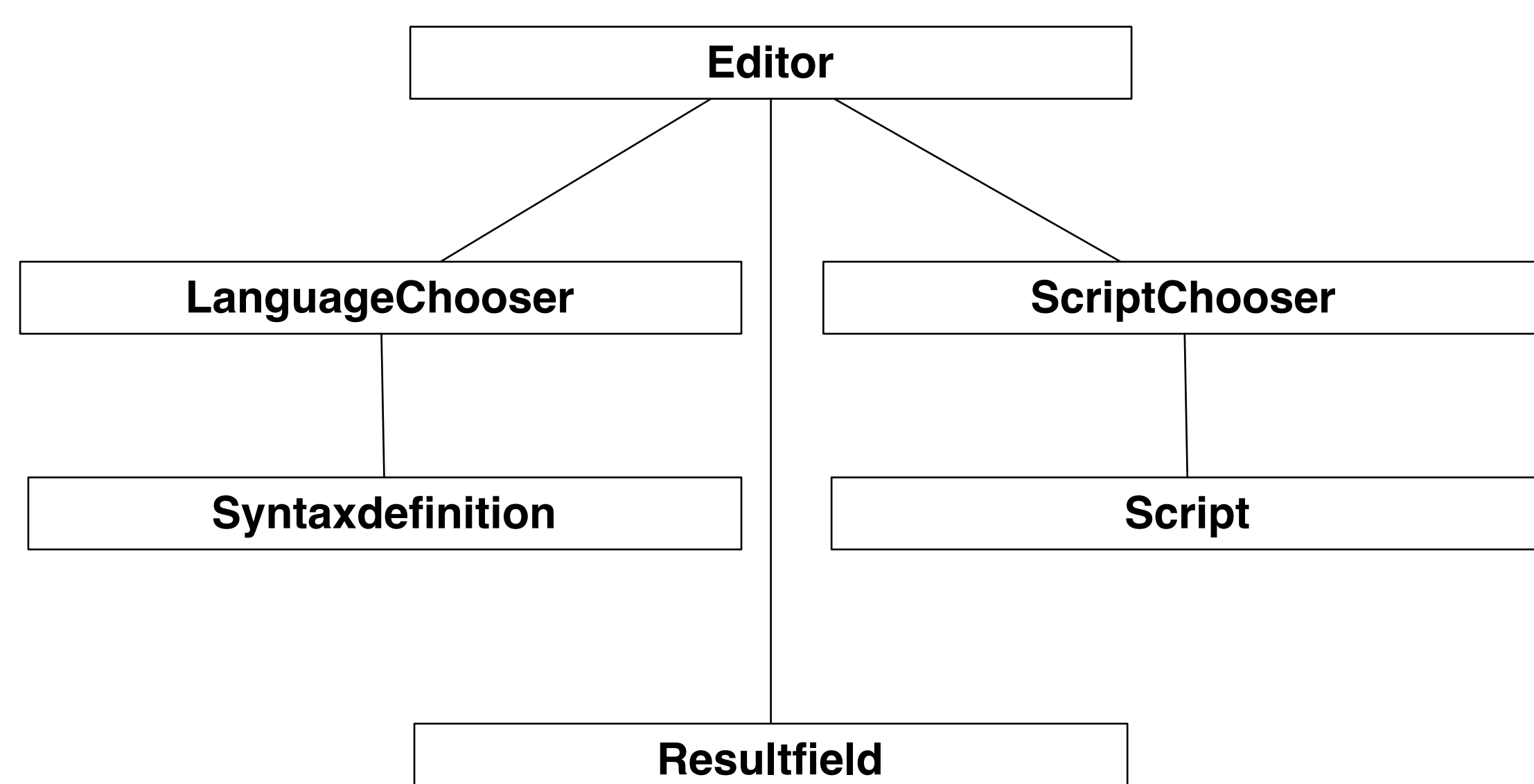
Language: Scripts:

```

1
2 proc dijkstra {graph origin} {
3   # Initialize
4   dict for {vertex distmap} $graph {
5     dict set dist $vertex Inf
6     dict set path $vertex {}
7   }
8   dict set dist $origin 0
9   dict set path $origin [list $origin]
10
11   while {[dict size $graph]} {
12     # Find unhandled node with least weight
13     set d Inf
14     dict for {uu -} $graph {
15       if {$d > [set dd [dict get $dist $uu]]} {
16         set u $uu
17         set d $dd
18       }
19     }
20   }
21 }

```

Architektur des Editors



Die Basis für den Editor bietet der Editor Ace, welcher die Grundfunktionalität zur Verfügung stellt. Die einzelnen Syntaxdefinitionen wurden im Zuge des Projektes entwickelt und können ähnlich wie die Skripte zur Laufzeit ausgewählt werden.

Evaluierung des Codes

- Ausführung in einem eigenen Interpreter im Safe mode
- spezielle Behandlung aller Ausgabebefehle um sie an den Editor zu retournieren
- Übermittlung der Zeile, in welcher der Fehler aufgetreten ist

Umsetzung

Christoph Hochreiner
E: h0726292@wu.ac.at

Betreuung

Stefan Sobernig
E: stefan.sobernig@wu.ac.at