

## Syndred - A Syntax-Driven Editor for Lexical Resources

### Mondaca, Francisco

f.mondaca@uni-koeln.de  
Universität zu Köln, Deutschland

### Rolshoven, Jürgen

rols@spinfo.uni-koeln.de  
Universität zu Köln, Deutschland

### Schildkamp, Philip

philip.schildkamp@uni-koeln.de  
Universität zu Köln, Deutschland

### Vogt, Andreas

vogt.andreas@uni-koeln.de  
Universität zu Köln, Deutschland

Kontinuierlich steigt die Menge der erzeugten und prozessierten Information und damit der Bedarf an technologisch assistierter bis hin zu autonomer Datenverarbeitung. Textuell repräsentierte Inhalte mittels strukturgebender Methoden aufzubereiten ist meist aufwendiger als deren Erzeugung, insofern gewinnt automatische Textprozessierung zunehmend an Bedeutung (Bernstein et al.: 2016).

Mithilfe der formalen Grammatiken RBNF (Rich Backus-Naur Form) <sup>1</sup> und ABNF (Augmented Backus-Naur Form) können Produktionssysteme im Sinne domänenspezifischer, formaler Sprachen (engl. Domain-Specific Languages) modelliert und auf textuelle Daten angewandt werden: Syndred <sup>2</sup> nutzt als Parser konkretisierte Grammatiken zur Abbildung von Texten auf Strukturbäume (engl. Parse Trees), die in vielfältiger Form (z. B. als TEI-Dokumente) persistiert, mittels Treewalkern (Parr 2013: 17ff) für weitere Prozessierung transformiert oder beispielhaft zur Überprüfung formaler Korrektheit unter Berücksichtigung von Merkmalen wie bspw. Schriftgröße, -stil und -farbe eingesetzt werden.

Wegweisend für die Entwicklung des hier beschriebenen Systems sind die Arbeiten von Wirth und Gutknecht (Gutknecht 1985 bzw. Wirth / Gutknecht 1992: 78ff). Entgegen der Funktionsweise eines Compilers jedoch, der die Analyse des zu übersetzenden Programms meist direkt in Programmcode vornimmt, definiert Syndred textuelle Strukturen anhand domänenspezifischer Grammatiken und bildet sie auf Syntaxgraphen ab.

Der syntaxkontrollierte Editor (engl. Syntax-Driven Editor, kurz: Syndred) soll als flexibles Werkzeug in

unterschiedlichsten Projekten (z. B. Lexikoneditoren und dezentraler Korrektur) dienen, kollaboratives Arbeiten durch effiziente formale Kontrolle unterstützen und damit die allgemeine Produktivität durch die Erstellung domänenspezifischer Sprachen erhöhen (Fowler 2011: xxi).

Syndred wartet mit einem Split-View Design auf, welches einerseits das Erstellen und Bearbeiten von Grammatiken und andererseits syntaxkontrolliertes Editieren textueller Inhalte in direkten, visueller Bezug zueinander bringt. Kernkompetenzen der Benutzerschnittstelle sind die Herleitungen von Grammatiken aus textuellen Bestandsdaten oder Rückmeldungen formaler Produktionsregeln zu textuellen Inhalten.

Dezentrales Arbeiten erfordert eine netzwerkbasierte Architektur, idealerweise eine Web-Applikation mit clientseitigem Editor und serverseitigem Parser; Syndred nutzt das auf ReactJS <sup>3</sup> basierende Editor-Framework DraftJS <sup>4</sup> und kommuniziert im JSON-Format über WebSockets mit einer zentralen, auf Basis des Spring-Frameworks <sup>5</sup> implementierten Java-Applikation. Neben dem Vorteil der Plattformunabhängigkeit ermöglichen die eingesetzten Web-Technologien die Zentralisierung der Hardwareleistung wie auch des Speicherbedarfs und sichern gegen Datenverlust ab. Auch garantiert die Bidirektionalität der Client-Server-Verbindung die Kohärenz und Persistenz der kollaborativen Instanzen. Zeitnahe Verfügbarkeit aller für die kollaborative Arbeit mit Syndred notwendigen Ressourcen wird durch einen serverseitigen Cached Thread Pool sichergestellt; aktiven Instanzen wird bei Bedarf ein Parser-Threads zur Verfügung gestellt, der nach Verwendung beendet wird.

Syndred ist somit ein Werkzeug zur intuitiven Entwicklung domänenspezifischer Sprachen und Überprüfung textueller Inhalte anhand dieser formalen Grammatiken, realisiert als kollaborative Web-Applikation; zusammengenommen ein Meilenstein in der Entwicklung dieser Art von Programmiersprachen.

## Fußnoten

1. RBNF ist eine EBNF (Extended Backus-Naur Form) Erweiterung um Rich-Text Auszeichnungen.
2. <https://github.com/spinfo/syndred>
3. <https://facebook.github.io/react>
4. <https://draftjs.org>
5. <https://projects.spring.io/spring-framework>

## Bibliographie

**Bernstein, Abraham / Hendler, James / Noy, Natalya** (2016): "A New Look at the Semantic Web." in: *Communications of the ACM*. 59:9. 35-37.

**Fowler, Martin** (2011): *Domain-Specific Languages*. Addison-Wesley: Upper Saddle River, New Jersey.

**Gutknecht, Jürg** (1985): "Concepts of the Text Editor Lara." in: *Communications of the ACM*. 28:9. 942-960.

**Parr, Terence** (2013): *The Definitive ANTLR 4 Reference*. Pragmatic Bookshelf: Dallas Texas.

**Wirth, Niklaus** (1986<sup>4</sup>): *Compilerbau*. Teubner: Stuttgart.

**Wirth, Niklaus** (2000<sup>5</sup>): *Algorithmen und Datenstrukturen*. Teubner: Stuttgart.

**Wirth, Niklaus / Gutknecht, Jürg** (1992): *Project Oberon: Design of an Operating System and Compiler*. Addison-Wesley Longman: Amsterdam.