technische universität dortmund

fakultät für informatik

DSL-Driven Generation of User Documentations for Web Applications

BACHELORARBEIT

MUKENDI MPUTU

JANUAR 12, 2022

Agenda

- Dokumentation von Webanwendungen
- Gegenstand der Arbeit
- Methodologie
- Ergebnisse
- Demo
- Fazit
- Referenzen

Quelle: https://www.today.com/home/men-or-women-whos-better-assembling-ikea-furniture-t59686

Dokumentation von Webanwendungen

Ziel und Relevanz

- Erleichtern das Erlernen der Bedienung der Anwendung mit Beschreibungen und Anleitungen
- Herausforderung durch regelmäßiges Updaten der Anwendung, vor allem wenn der Benutzer der Aktualisierung nicht zustimmen kann

- Gut strukturierte Dokumentation profitiert nicht nur dem User, sondern der entwickelnden Firma auch

Dokumentation von Webanwendungen

Ziel und Relevanz

- Erster Ansatzpunkt für den Entwickler dem Endbenutzer das Wissen zu vermitteln, um mit der Anwendung angemessen zu interagieren
- In unserem Fall ▸ Beschreibung und Navigationsanweisungen der User Interface (UI) mit unterstützenden Bildern
- Normiertes Vorgehen zur Dokumentation durch ISO/IEEE (z.B. ISO-26514 oder -26511)

Gegenstand der Arbeit

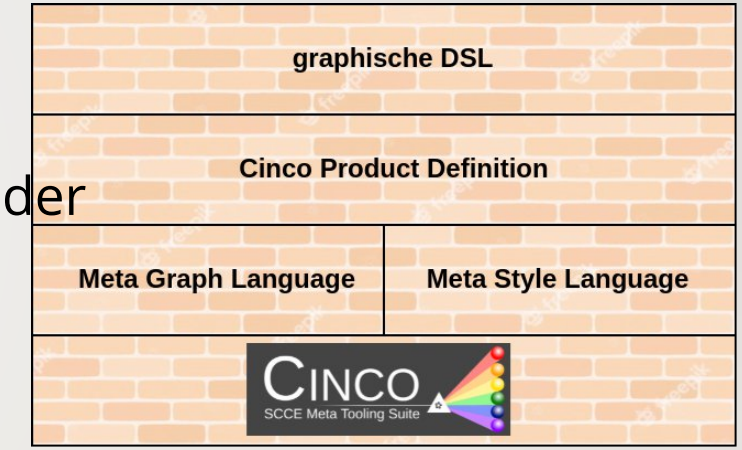
Eine DSL zur automatischen Generation der Nutzerdokumentation

- Model-basierte Lösung vorschlagen, die die Vorteile der Entwicklung mit DSL ausnutzt, um Endbenutzerdokumentation für Webanwendungen automatisch zu generieren
- DSLs erlauben eine Repräsentation des Problems (des Systems) zunächst als Model, um dann daraus eine Lösung zu generieren
- Modelle der Dokumentation zum aktuellen Stand der Anwendung können auch Nicht-Programmierer erstellen

Methodologie

*graphische DSL aus **Cinco** Metasprachen*

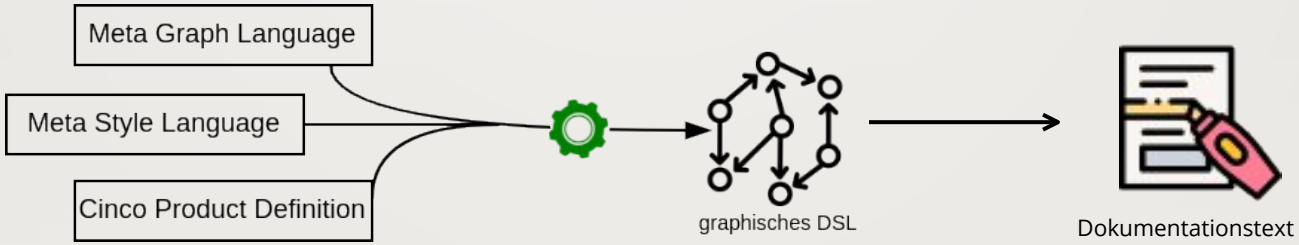
- Den Grundstein liefert die Cinco Entwicklungsumgebung samt der Modellierungssprachen
- Diese Modellierungssprachen fundieren als Meta-Modellierungssprachen zur graphischen Sprache mit der Dokumentationsmodelle entworfen werden



Methodologie

*graphische DSL aus **Cinco** Metasprachen*

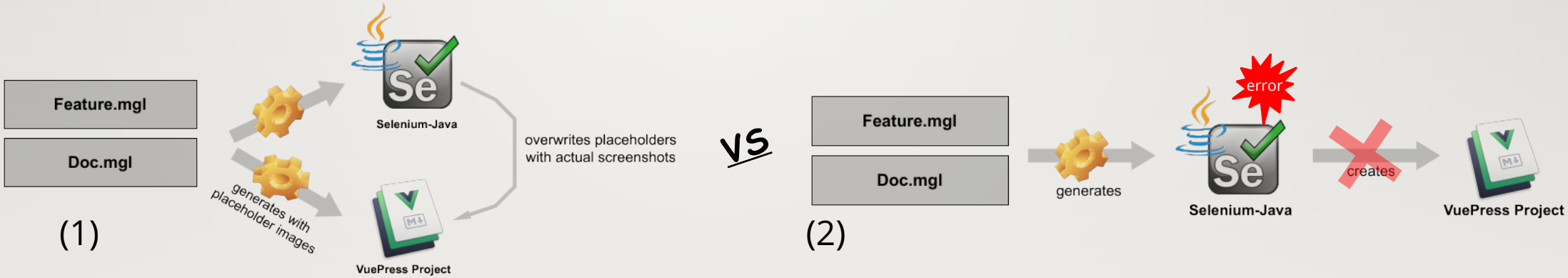
- Cinco Metasprachen (Meta Graph Language, Meta Style Language, Cinco Product Definition) spezifizieren unsere graphische Domainsprache
- Diese graphische Sprache - bestehend aus Repräsentationen der Browser UI Elemente - wird zum Entwerfen des Dokumentationsmodels verwendet
- Das Endresultat ist eine Dokumentation in Textformat (Markdown) mit Screenshot



Methodologie

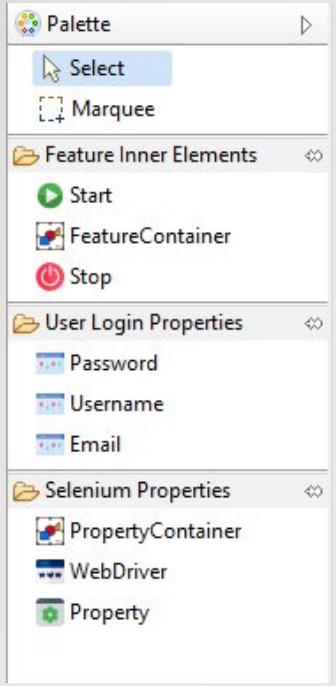
*graphische DSL aus **Cinco** Metasprachen - technisches Detail*

- Verwendung von Selenium-Java und VuePress, um einen Dokumentationstext mit Bildern in Markdownformat zu erstellen
- Separates Erzeugen der Texte und der Screenshots (1) wird bevorzugt, um Single-Point-of-failure zu vermeiden, im Falle eines Fehler im Vorprozess (2)



Graphische DSL: .feat und .doc

FeatureGraph Modellsprache



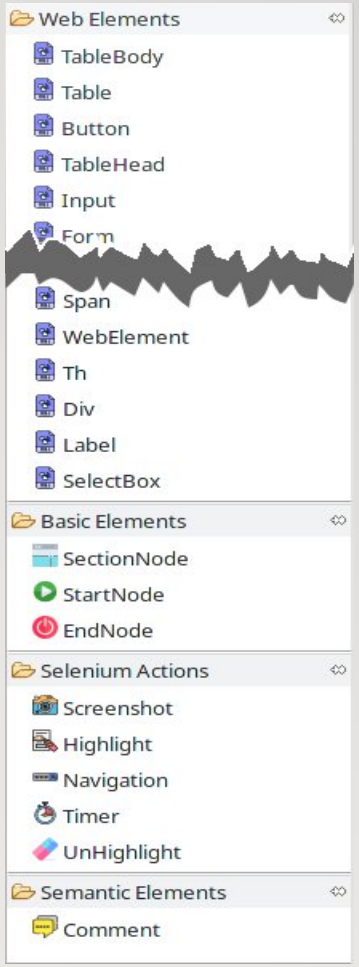
FeatureGraph Modellelemente

- Modellelement sind nach ihrem Zweck kategorisiert
- ‚Selenium Properites' z.B. kategorisiert Einstellungselemente für die Selenium Framework, d.h. Konfigurationen, die sonst nicht modelliert werden könnten, aber dennoch für die Ausführung der Anwendung wichtig sind
- ‚Feature Inner Elements' repräsentieren Strukturelemente der Modellsprache .feat

Graphische DSL: .feat und .doc

DocGraph Modellsprache

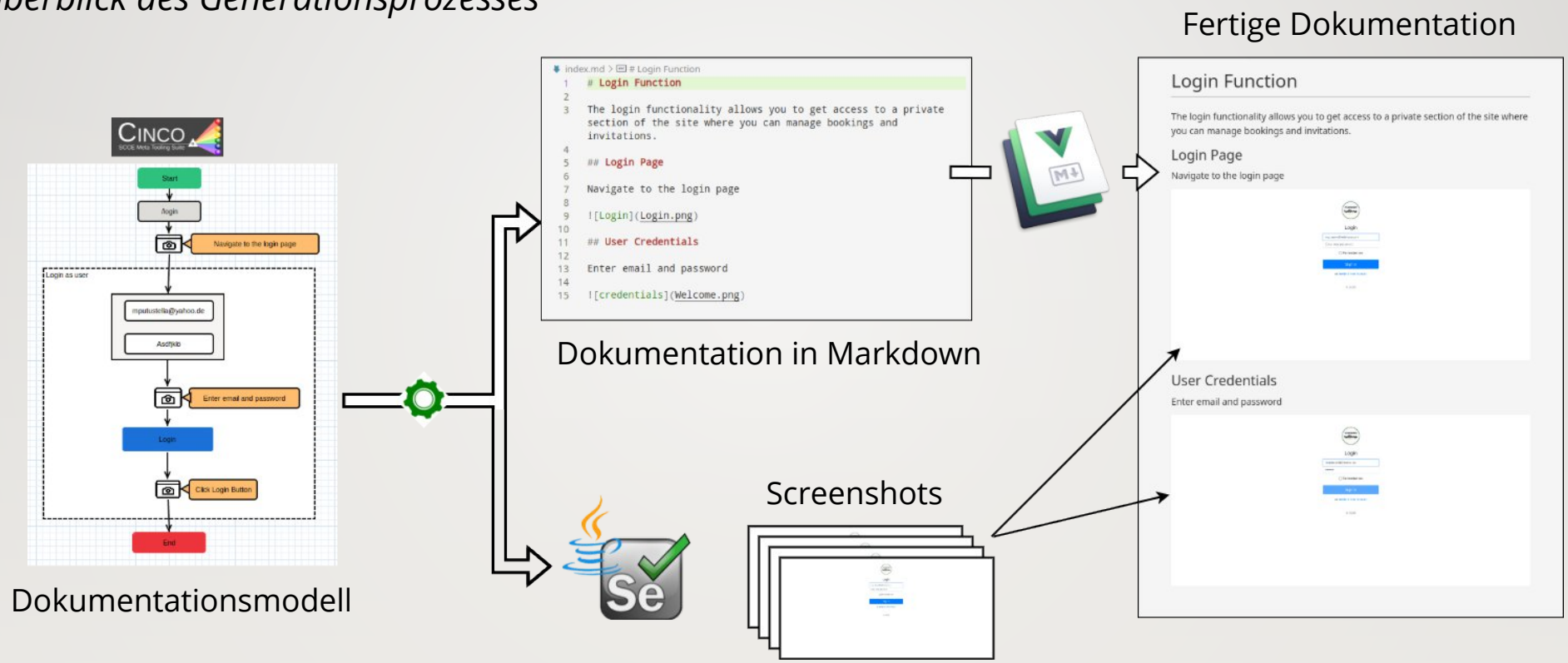
- DocGraph Modellelemente sind hauptsächlich ‚Web Elements‘; Repräsentationen von HTML-Elemente mit Schlüsseleigenschaften (wie Selektoren, ids, usw.)
- ‚Selenium Actions‘ sind Elemente, die Methoden des Selenium WebDrivers specifizieren, welche den Browser antreiben
- Die Semantik der Modellobjekte werden mit dem Comment-Element oder mit der Description-Eigenschaft der einzelnen Elemente definiert



DocGraph Modellelemente

Graphische DSL: .feat und .doc

Überblick des Generationsprozesses



Ergebnisse

WebDoc - Web Application Documentor

- Modell-basierter Editor zum Erstellen von Nutzerdokumentation für Webanwendungen
- Zwei Modellsprachen, die jeweils die Funktionalitäten und die darin enthaltene Nutzeraktionen beschreiben
- Ein Modell-zu-Text-Generator, der Markdowndateien mit Dummybildern erstellt

Ergebnisse

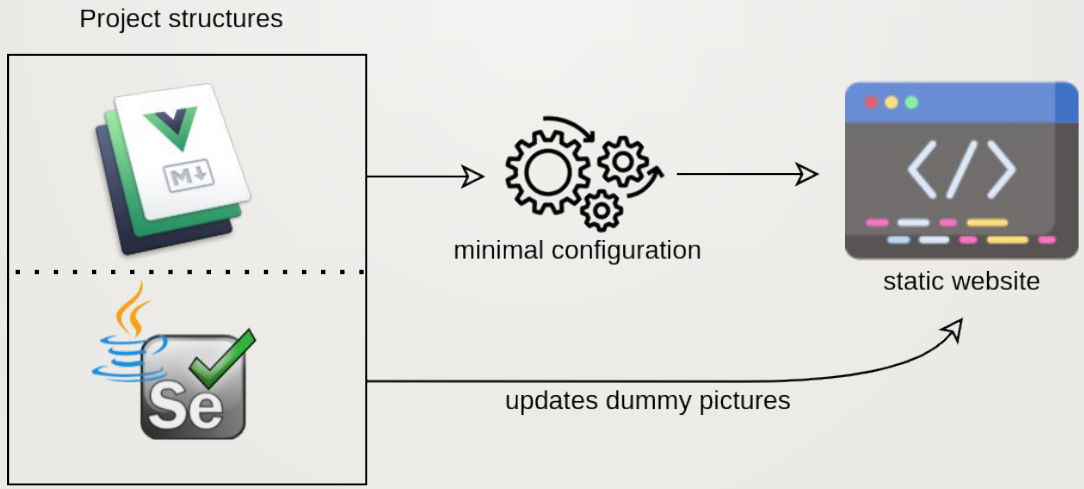
WebDoc - Web Application Documentor

- Implementierung der Wiederverwendbarkeit der Modellelemente, um den Designprozess zu beschleunigen
- Überprüfung der Korrektheit des Graphmodells vor der Generation, um Fehler zu minimieren
- Generierung der gesamten Projektstruktur mit ‚Boilerplate-code' innerhalb desselben Projektordners

Ergebnisse

WebDoc - Web Application Documentor

- Ad hoc Erstellung eine Dokumentations-Webseite durch Integration von Drittanbierter-Tools, VuePress und Markdown
- Starten der Dokumentations-Website mit minimalem Konfigurationsaufwand



technische universität dortmund

fakultät für informatik

Demo

WebDoc - Web Application Documentor

Künftige Arbeiten

- Erleichterung des Erstellungsprozesses, indem eine Projektvorlage mit einer Startkonfiguration erstellt wird
- Querverweis auf DocGraphModels innerhalb anderer könnte in Zukunft verbessert werden, um beispielsweise alle verfügbaren Graphenmodelle in einer separaten Ansicht auflisten zu können
- Aufbesserung der graphischen Sprachen mit weiteren Selektoren, um HTML-Elemente besser addressieren zu können
- Weitere Checks implementieren, um beispielsweise die Syntax der Selektoren im Voraus zu validieren

Fazit

- Funktionalität: DSL-getriebene Generierung von Endbenutzerdokumentation für Webanwendungen auf der Basis von Graphenmodellen unter Verwendung einer grafischen DSL
- Benutzerfreundlichkeit: Die vorgestellten Sprachfeature ermöglichen einen intuitiven Entwurf eines Dokumentationsmodells, während verschiedene Checks helfen, syntaktisch korrekte Graphmodelle zu erstellen
- Leistung: schnelle Erstellung einer Dokumentationswebsite mit minimaler Konfigurationsaufwand durch Integration bewährter Drittanbieter-Tools

Vielen Dank!