Abstract

Automatisierung war schon immer die beste Lösung, um mit großen Massen von Dateien umzugehen(bearbeiten? verarbeiten?), indem neue Ideen eingeführt werden, um die endlosen Analyseprozessezeit (Analyse-aufwand? verringern) zu verkürzen.
Die Hauptidee der Bachelorarbeit besteht darin, Xml , Graphen und deren Technicken zu untersuchen. Das Ziel dieser Arbeit ist es, große Massen von XML-Datein in Graphen durch ein Tool umzuwandeln, um dem Team, das in wossidia arbeitet, dabei zu unterstützen, Bei Analysieren, Durchsuchen und Visualisieren bestimmter Aspekte in den gesammelten Daten.
Um dieses Ziel zu erreichen, untersuche und analysiere ich die XML- und Graphdateien, die sich auf den Konvertierungsalgorithmus beziehen.
Der Code des Tools wird in Java geschrieben und verwendet verschiedene hilfreiche Bibliotheken wie z.B. XPath bzw XPathFactory, org.w3c, org.xml ...
\paragraph {Schritt A: Filter} //fur latex hier
Wir haben viele XML-Dateien, die unterschiedliche Inhalte enthalten, und diese Dateien werden nach den Angaben des Benutzers gefiltert, d. h. der Benutzer kann entscheiden, welche Dateien mit welchen spezifischen Inhalt verarbeiten werden können.
\paragraph {Schritt B: Transformator} //fur latex auch
Die Transformation sollte auch von benutzerdefinierten Regeln geleitet werden, die eine Abbildung von XML-Schema-Konzepten auf Property-Graph-Modellkonzepte beschreiben. Dadurch sollte der Ethnologe und Forscher durch eine sogenannte Regelspezifikationssprache in der Lage sein, Regeln zu definieren, die Elemente, Attribute und Inhalte aus XML-Dateien auswählen und aus dieser Auswahl Knoten, Labels, Kanten und Eigenschaften eines Graphen generieren.

Einführung

Viele interessante Programmierformalismen beschäftigen sich explizit mit XML-Dokumenten. Die Beispiele reichen von domain-specific

languages wie XSLT und XQuery bis hin zu general-purpose languages wie Java, in denen XML-Dokumente von speziellen Frameworks oder einfach als Text verarbeitet werden können.

Wenn solche Programme subject statischer Analysen sind, ist es notwendig, ein formales Modell von sets von XML-Dokumenten oder -Fragmenten zu erhalten, typischerweise um konservative Annäherungen an die möglichen Ergebnisse an bestimmten Programmpunkten darzustellen. Mehrere solcher Modelle wurden vorgeschlagen, hauptsächlich basierend auf der Beobachtung, dass formale Baumsprachen viele gewünschte Eigenschaften erfassen, da XML-Dokumente im Wesentlichen Bäume sind[1, 2]. Für die praktische Anwendung muss ein gutes und vielseitiges Modell für die statische Analyse einige besondere Anforderungen erfüllen:

• es muss alle für die Validierung relevanten Merkmale in XML erfassen, nicht nur eine idealisierte Teilmenge – insbesondere dürfen wir Attribute, Zeichendaten oder verschachtelte Inhaltsmodelle nicht ignorieren;

• es muss in der Lage sein, Sätze von XML-Dokumenten auszudrücken, die durch gängige Schemaformalismen beschrieben werden, insbesondere XML-Schema[3]

• es muss eine statische Validierung gegen gängige Schemaformalismen und auch eine Navigation mit XPath-Ausdrücken ermöglichen[4]

• es muss eine Gitterstruktur endlicher Höhe zur Verwendung in der Datenflussanalyse mit Festkomma-Iteration bereitstellen; und

• es muss vollständig umgesetzt werden.

In diesem Arbeit beschreiben wir das X2Gmodell, das all diese Kriterien erfüllt. wir haben dieses Programm mit Java geschrieben.

Es wurde in umfangreichen praktischen Szenarien beim Erstellen der Graphen von Wossidai-XML-Daten verwendet/ausprobiert/umgesetzt

XML2Graphen sind vollständig implementiert und in einem Github verfügbar.

[1]James Clark and Steve DeRose. XML path language, November 1999. W3C Recommendation. http://www.w3.org/TR/xpath.

[2] Haruo Hosoya, Jerome Vouillon, and Benjamin C. Pierce. Regular expression types for XML. ACM Transactions on Programming Languages and Systems, 27(1):46–90, 2005.

[3] Makoto Murata, Dongwon Lee, Murali Mani, and Kohsuke Kawaguchi. Taxonomy of XML schema languages using formal language theory. ACM Transactions on Internet Technology, 5(4):660–704, 2005.

[4] Henry S. Thompson, David Beech, Murray Maloney, and Noah Mendelsohn. XML Schema part 1: Structures second edition, October 2004. W3C Recommendation. http://www.w3.org/TR/xmlschema-1/.

Motivation

Forscher, die einen Massiven mengen von XML-Dateien haben, haben große Schwierigkeiten, mit dem Analysieren sowie mit Durchsuchen der Dateien, was im normalen fall sehr lange dauert. Dadurch wird die effeizenz der Arbeit groß eingeschränkt.

X2G unterstützt die Forscher diese schwierigkeit zu überwinden, indem alle gesammelten XML-Daten in Property Graph-Daten umzuwandeln.

X2G ist daher ein interessantes Projekt nicht nur wegen der Möglichkeiten, die es den Forschern in der Bibliothek zur Datenanalyse zur Verfügung stellt, sondern auch wegen seiner ganzheitlichen Perspektive, da sein Hauptzweck darin besteht, eine allgemeine Methodik zu sein, die es dem Benutzer ermöglicht, Regeln zu definieren die Elemente, Attribute und Inhalte aus XML-Dateien auswählen und aus dieser Auswahl Knoten, Labels, Kanten und

Eigenschaften eines Graphen generieren, um zur WeiterEntwicklung dieses Technologiefelds beizutragen.

Problemstellung

Wie bereits erwähnt, ist X2G als Konvertierungstool konzipiert. Daher muss es in der Lage sein, die Eingaben des Benutzers zu berücksichtigen, um den Graphen zu erstellen. Gemeint ist hier: welche Knoten werden mit welchen anderen Knoten verbunden

Ein möglicher Weg, dieses Ziel zu erreichen, wird in dieser Bachelorarbeit anhand bestimmter Bibliotheken in Java vorgestellt, mit denen ein Benutzer einen Graphen mit verschiedenen Eingaben erstellen kann.

Basierend auf persönlichen Untersuchung benötigen wir hauptsächlich bestimmte Texte wie Story (in DEU, ENG, NDS), Titel der Story, Orte, Personeninformationen etc ... wenn wir sie mit dem Tool in der xml-Datei finden können.

Graphen werden durch bestimmten Prozess erstellt, zuerst müssen wir die vom Benutzer ausgewählten Knoten und auch die Kanten in .CSV-Dateien wie Nodes.csv und Edges.csv speichern und dann den Graphen aus den .csv-Dateien erstellen

---

Aufgabeneinstellung

Die Hauptaufgabe dieser Arbeit besteht darin, mit dem X2G-Tool eine Transformation (XML TO GRAPH) zu implementieren, die es den Forschern ermöglicht, bestimmte Elemente aus den xml-Dateien mit xpath-Ausdrucken auszuwählen.

Dadurch wird die Transformation genauer. Dazu sind zwei Hauptschritte definiert. Zuerst (Filter) werden die erforderlichen Elemente der gegebenen Benutzereingabe mithilfe von Java-Bibliotheken Xpath extrahiert. Zweitens (Transformator) diese Elemente werden zu Nodes.

Diese beiden Schritte werden durch die Verwendung dieser Algorithmen in Java erreicht:

• XPath steht für XML Path Language [5]

• xfact
hier kommt noch was* in diesem Abschnitt esist noch nicht vollständig

[5]https://www.w3.org/TR/xpath-31/

Methodik

Das Ergebnis des vorgestellten Projekts ist ein werkzeug, das die XML-Dateien mit den Java-Bibliotheken (**javax.xml.xpath and.parsers, org.w3c.,java.io.FileWriter, Java.io.File, java.utility.ArrayList, java.util.HashMap** ….usw.) in Graphen umwandelt. Mit diesen Komponenten bildet die Arbeit die Grundlage für ein Werkzeug, das in vielen weiteren Szenarien der XLM-GRAPH-Transformation eingesetzt werden kann.

Übersicht zur XML, Graphen und deren Technicken state of

Was ist und warum XML?[6]

XML steht für Extensible Markup Language. Markup Language ist ein Satz von Codes oder Tags, die den Text in einem digitalen Dokument beschreiben.

Die bekannteste Auszeichnungssprache ist die Hypertext Markup Language (HTML), die zum Formatieren von Webseiten verwendet wird. XML, ein flexiblerer Verwandter von HTML, ermöglicht die Abwicklung komplexer business über das Internet.

Die Flexibilität von XML hat viele Vorteile.

Damit können Sie Daten zwischen Unternehmensdatenbanken und Websites übertragen, ohne wichtige beschreibende Informationen zu verlieren.

Sie können die Präsentation der Daten automatisch anpassen, anstatt allen Teilnehmern dieselbe Seite anzuzeigen.

Und es macht die Suche effizienter, da Suchmaschinen präzise Tags anstelle langer Textseiten sortieren können.

---

[6]https://hbr.org/2000/07/explaining-xml

Wofür wird XML verwendet?[7]

XML ist heute eines der am häufigsten verwendeten Formate für den Austausch strukturierter Informationen: zwischen Programmen, zwischen Menschen, zwischen Computern und Menschen, sowohl lokal als auch über Netzwerke.

Ein kurzes Beispiel:

```
<part number="1976">
  <name>Windscreen Wiper</name>
  <description>The Windscreen wiper
    automatically removes rain
    from your windscreen, if it
    should happen to splash there.
    It has a rubber <ref part="1977">blade</ref>
    which can be ordered separately
    if you need to replace it.
```

```
  </description>
</part>
>
```

Wenn man bereits mit HTML vertraut ist, kann man feststellen, dass XML sehr ähnlich ist. Die Syntaxregeln von XML sind jedoch streng: XML-Tools verarbeiten keine Dateien, die Fehler enthalten, sondern geben uns Fehlermeldungen, damit wir diese beheben können. Das bedeutet, dass fast alle XML-Dokumente zuverlässig von Computersoftware verarbeitet werden können.

[7] https://www.w3.org/standards/xml/core

Warum sind Bibliothekare so begeistert von XML?[8]

XML ist wichtig, weil es das Teilen und Durchsuchen von Ressourcen in verschiedenen Formaten viel einfacher macht

Bis vor kurzem war dies für Bibliotheken kein großes Problem. Historisch gesehen haben Bibliotheken als zentrale Informationsspeicher gedient.

Sie kauften Bücher, Zeitschriften, Filme und andere Informationsressourcen auf physischen Medien, und die Benutzer fanden den Besitz der Bibliothek in einem Katalog, der die Bestände auflistete.

Die meisten Kataloge werden unter der Annahme entworfen, dass, sobald eine Bibliothek einige beschreibende Informationen zu jeder von ihr gekauften Ressource erfasst, diese Informationen nicht radikal geändert werden müssen.

Bei physischen Ressourcen funktioniert dies ziemlich gut, da sich die Autoren, Titel, Themen und physischen Eigenschaften eines Buches nicht ändern.

Als der Zugang zum Internet weit verbreitet war, wurde klar, dass die Bereitstellung des Zugangs zu entfernten elektronischen Ressourcen sehr problematisch sein kann.

Kataloge sollen den Zugriff auf physische Ressourcen ermöglichen, die der direkten Kontrolle der Bibliothek unterliegen.

Die Leute möchten jedoch Zeitschriftenartikel, Bücher und nützliche Webseiten lesen, die in dynamisch aktualisierten Datenbanken gespeichert sind, die von anderen Organisationen gepflegt und verwaltet werden, die möglicherweise Tausende von Kilometern entfernt sind.

Online-Bibliothekskataloge sind für den Zugriff auf diese Werke schlecht geeignet, daher nehmen viele Bibliotheken diese Art von Ressourcen nicht in den Katalog auf.

Infolgedessen ist es für Benutzer oft sehr schwierig zu wissen, welche elektronischen Ressourcen sie über ihre Bibliotheken erhalten können.

Hier kommt XML ins Spiel. Es ist unmöglich, Informationen zu suchen oder anzuzeigen, wenn sie nicht sinnvoll strukturiert sind.

Im Klartext bedeutet dies, dass sich Informationsanbieter auf Standards für die Kodierung elektronischer Dokumente einigen müssen, damit diese einheitlich abgerufen werden können.

Bibliotheken haben seit vielen Jahren bibliografische Datensätze in MARC kodiert, was es ihnen ermöglicht, Katalogdatensätze einfach gemeinsam zu nutzen, was Kosten senkt und gleichzeitig den Service verbessert.

Aus verschiedenen Gründen ist es nicht möglich, die neuen Arten von Ressourcen, auf die Benutzer zugreifen möchten, in MARC zu codieren. Wenn die Informationen jedoch in XML gespeichert werden, ist es möglich, diese Daten auf eine Weise zu teilen und zu kombinieren, die sonst nicht möglich wäre.

[8] https://www.infotoday.com/cilmag/sep02/Banerjee.htm

Was ist und warum Graphen?[9]

Graphen und Diagramme sind effektive visuelle Werkzeuge, da sie Informationen schnell und einfach darstellen. Es überrascht daher nicht, dass Graphen häufig von Print- und elektronischen Medien verwendet werden. Manchmal können Daten besser verstanden werden, wenn sie in einem Diagramm dargestellt werden als in einer Tabelle, da das Diagramm einen Trend oder einen Vergleich aufzeigen kann.

Für die Darstellung von Statistiken ist es wichtig zu wissen, wie man Informationen grafisch vermittelt. Im Folgenden finden Sie eine Liste mit allgemeinen Regeln, die Sie bei der Erstellung von Graphen beachten sollten.

Eine gute Graph:

• stellt die Fakten genau dar,

• erregt die Aufmerksamkeit des Lesers,

• ergänzt oder demonstriert Argumente im Text,

• hat einen Titel und Labels,

• einfach und übersichtlich ist,

• zeigt Daten an, ohne die Nachricht der Daten zu ändern,

• zeigt deutlich alle Trends oder Unterschiede in den Daten,

• visuell korrekt ist (d. h. wenn ein Diagrammwert 15 und ein anderer 30 beträgt, sollte 30 doppelt so groß wie 15 erscheinen).

Warum Graphen zur Darstellung von Daten verwenden?

Weil sie…

• sind schnell und direkt,

• die wichtigsten Fakten hervorheben,

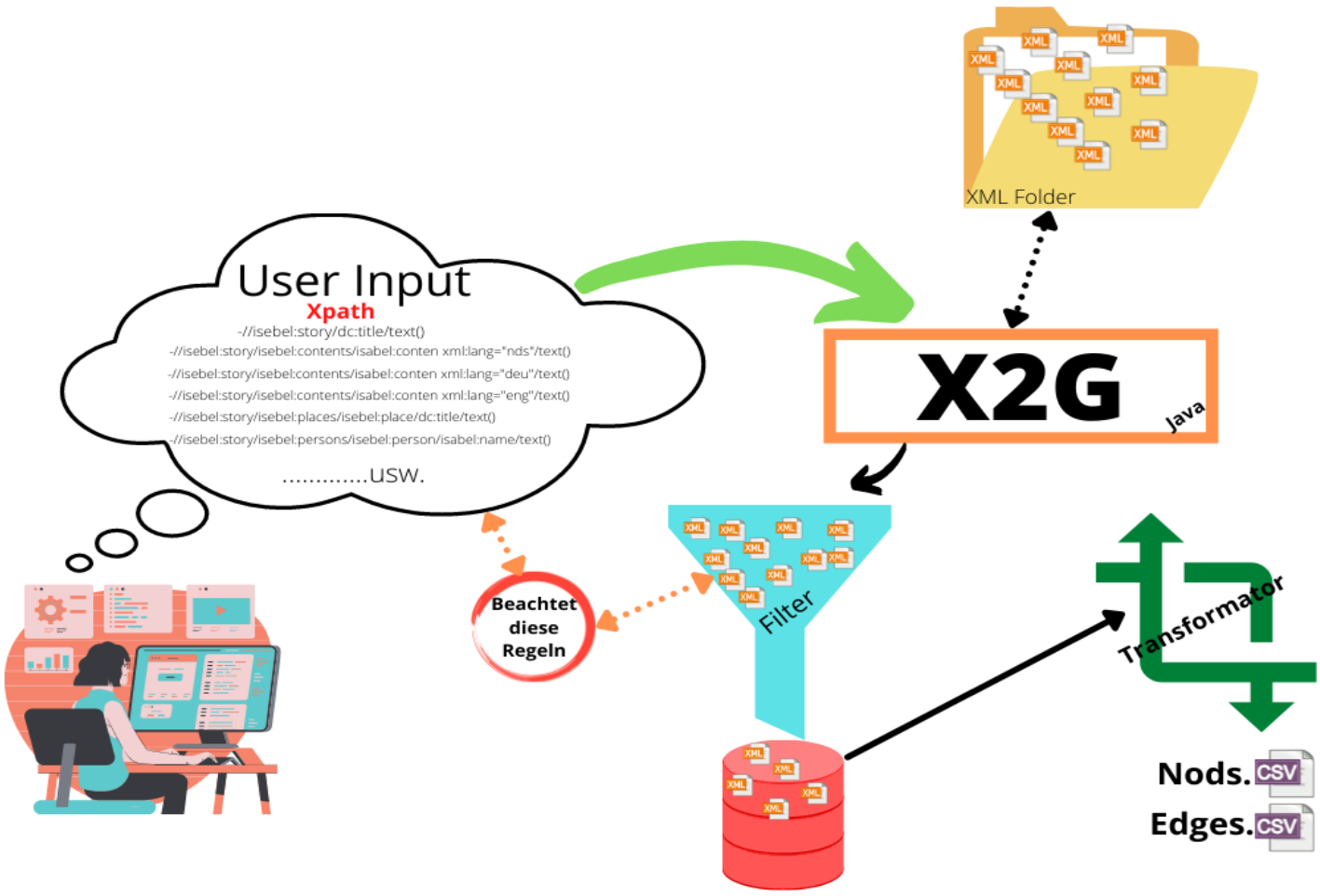
• das Verständnis der Daten erleichtern,

• kann die Leser überzeugen,

• lässt sich leicht merken.

[9] https://www150.statcan.gc.ca/n1/edu/power-pouvoir/ch9/5214821-eng.htm

Konzept



Das bild wird noch detalierter also (Die funktionsweise fur jeden schritt) kommt noch

---

XPath steht für XML Path Language

XPath verwendet eine "pfadähnliche" Syntax, um Knoten in einem XML-Dokument zu identifizieren und zu navigieren

XPath enthält über 200 integrierte Funktionen

XPath ist ein wichtiges Element des XSLT-Standards

XPath verwendet Pfadausdrücke, um Knoten oder Knotengruppen in einem XML-Dokument auszuwählen.

Diese Pfadausdrücke sehen den Pfadausdrücken sehr ähnlich, die Sie mit herkömmlichen Computerdateisystemen verwenden:

XPath-Standardfunktionen

XPath enthält über 200 integrierte Funktionen.

Es gibt Funktionen für Zeichenfolgenwerte, numerische Werte, Boolesche Werte, Datums- und Zeitvergleich, Knotenmanipulation, Sequenzmanipulation und vieles mehr.

Heute können XPath-Ausdrücke auch in JavaScript, Java, XML-Schema, PHP, Python, C und C++ und vielen anderen Sprachen verwendet werden.

XPath-Terminologie

Knoten

In XPath gibt es sieben Arten von Knoten: Element-, Attribut-, Text-, Namensraum-, Verarbeitungsanweisungs-, Kommentar- und Dokumentknoten.

XML-Dokumente werden als Knotenbäume behandelt. Das oberste Element des Baums wird als Wurzelelement bezeichnet.

Wie sehen die Isabel daten aus?