XSTEP-ENTWICKLERTAGUNG HDM STUTTGART 7./8. MAI 2010

Teilnehmer:

Anne Auditor (Universität Tübingen)
Irmela Bauer-Klöden (Universität Tübingen)
Björn Dünckel (pagina GmbH)
Oliver Gasperlin (pagina GmbH)
Peter Gietz (DAASI International)
Marko Hedler (Hochschule der Medien Stuttgart)
Thomas Kollatz (Uni Duisburg)
Stephan Moser (Universität Würzburg)
Matthias Osthof (Universität Zürich)
Tobias Ott (Hochschule der Medien Stuttgart)
Hannelore Ott (pagina GmbH)
Wilhelm Ott (Universität Tübingen)
Ute Recker-Hamm (Universität Trier)
Kuno Schälkle (Universität Tübingen)
Heino Schmull (pagina GmbH)
Michael Trauth (Universität Trier)
Gabriel Viehhauser (Universität Bern)

AUSGANGSSITUATION UND PROJEKTZIEL

Das Programmpaket TUSTEP wird seit über 35 Jahren am Rechenzentrum der Universität Tübingen entwickelt und gepflegt. TUSTEP ist eine Skriptsprache und ein Satzsystem für die Anwendung vor allem in den Geisteswissenschaften und ist im philologischen Umfeld bis heute unerreicht in Hinblick auf Leistungsumfang, Performanz und Flexibilität. TUSTEP wendet sich dabei v.a. an Forschungsbereiche, bei denen die Texte selbst Gegenstand der Untersuchungen sind, also z.B. sprachwissenschaftliche Analysen, statistische Auswertungen, Textvergleiche, linguistische Untersuchungen etc. Das angeschlossene Satzsystem ist mit derzeit ca. 500.000 Seiten Umbruch/ Minute auf einem handelsüblichen Arbeitsplatz-Rechner immer noch das mit Abstand performanteste Umbruchsystem der Welt.
Weltweit wird TUSTEP an über 60 Universitäten und Forschungseinrichtungen eingesetzt.
Nun leidet die Akzeptanz von TUSTEP aber an einer Sache: Die Syntax der TUSTEP-Proramme ist

seit 35 Jahren gewachsen, ist proprietär, nicht intuitiv, gilt vielen als schwer zu erlernen und nicht mehr zeitgemäß - der philologische Nachwuchs behilft sich daher häufig lieber mit weit weniger leistungsfähigen, aber besser bedienbaren Tools wie z.B. Perl.

Eine zentrale Eigenschaft von TUSTEP ist, mit (fast) jeder Form von Textdaten umgehen zu können, also nicht auf XML-strukturierte Daten angewiesen zu sein. Damit ist TUSTEP auch für ein Arbeitsfeld prädestiniert, das noch immer nicht befriedigend besetzt ist, nämlich aus den verschiedensten Quelldaten skriptbasiert XML-Daten zu erzeugen.

Tobias Ott, wiss. MA an der Hochschule der Medien in Stuttgart und Geschäftsführer der Firma pagina GmbH in Tübingen, hat im vergangenen Jahr eine Konzeptidee entwickelt, die TUSTEP-Syntax in XML abzubilden, um den vollen Leistungsumfang dieses Programmpaketes in einer modernen, zum Selbststudium geeigneten Umgebung anbieten zu können. Die Vorteile einer solchen XML-basierten Syntax liegen auf der Hand: Unterstützung eines offenen Standards, weite Verbreitung, Programmierung in jedem XML-Editor, automatische Syntaxprüfung, Code Completion und klare Schnittstellen sind nur die naheliegendsten Argumente für diese Arbeit. Die Tatsachen, dass der eigentliche Programmkern (der „Prozessor“) wohl nicht oder kaum verändert werden muss und dass TUSTEP mittlerweile Open Source ist, befördern das Vorhaben ebenfalls.

Die Vorstellung dieser Idee auf zwei Tagungen im Januar diesen Jahres in Blaubeuren und Trier ergab ein mehr als eindeutiges Stimmungsbild: Sämtliche Teilnehmer unterstützen das Vorhaben und viele haben auch ihre Mitarbeit zugesagt. Darunter ist neben den erfahrenen TUSTEP-Anwendern auch die Forschungsinitiative TextGrid (http://www.textgrid.de/), die eine Chance darin sieht, die über das TextGridLab zugänglichen Leistungen mit TUSTEP-Skripten wesentlich zu erweitern. Eine Umsetzung der TUSTEP-Syntax nach XML würde die Integration dramatisch erleichtern.

Am 7. und 8. Mai 2010 fand auf Einladung von Prof. Dr. Marko Hedler und Dipl.Ing Tobias Ott (beide HdM) und Prof. Dr. Wilhelm Ott (Universität Tübingen) das erste offizielle Treffen von Interessenten für die Umstellung der TUSTEP-Syntax nach XML (Arbeitstitel: XSTEP) an der HdM statt. Bei dem Gründungstreffen ging es sowohl um technische als auch um organisatorische Fragen.

Prof. Dr. Alexander Roos, Rektor der Hochschule der Medien, begrüßt die Teilnehmer und stellt die Hochschule als Europas größte Medienhochschule mit rund 20 Studiengängen zur Medienbranche vor. Er begrüßte die Initiative, die sich gut in das Forschungsumfeld der HdM einfüge.

Die Tagung begann mit zwei Eröffnungsvorträgen von Tobias Ott zum Konzept von XSTEP und Prof. Dr. Marko Hedler zur Arbeitsweise von XML-basierten Skriptsprachen,

Im Folgenden sollen stichpunktartig die wichtigsten Ergebnisse der Tagung aufgelistet werden.

TEIL 1: ALLGEMEINES KONZEPT

- **Formulierung des Ziels:** Schaffung einer eigenen Skriptsprache XSTEP, die – wie etwa auch XSLT – selbst wieder in XML geschrieben ist und möglichst den vollen Funktionsumfang von TUSTEP abbildet. Dabei soll nicht der eigentliche Programmkern von TUSTEP verändert werden - geschaffen wird vielmehr ein "XML-basiertes Frontend" für TUSTEP, während TUSTEP selbst weiterhin als Prozessor fungiert (analog z.B. zu Saxon für XSLT).
  XSTEP schafft somit eine Alternative zur Programmierung von TUSTEP-Skripten in der TUSTEP-eigenen Syntax (i.d.R. im TUSTEP-eigenen Editor). Ein XSTEP-Skript beinhaltet somit die Aufrufe der parametergesteuerten Programm-Module sowie notwendige Möglichkeiten der Ablaufsteuerung mit Schleifen, Bedingungen etc. Das Skript wird im Ganzen an den TUSTEP-Prozessor über eine noch zu definierende Schnittstelle übergeben und dort abgearbeitet.

- **Vorteile**:
  - Die Skripte können in einem gängigen XML-Editor geschrieben werden (z.B. Oxygen)
  - Syntaxprüfung erfolgt schon beim Verfassen über XML-Schema, verpflichtende Teile eines Skripts können automatisch aufgebaut werden.
  - CSS-basierte, selbsterklärende Oberfläche im Autormodus wird möglich
  - Automatische Code Completion
  - Über eine Anpassung des Editors (beim Oxygen: sog. Framework = offene Schnittstelle zum Oxygen-Editor) können umfangreiche Hilfen zu Erstellung der Skripte gegeben werden:
  - dadurch autodidaktisch erlernbar
  - leicht einzubinden in XML-Umgebungen (z.B. TextGrid)

- **Zu lösende Probleme:**
  - Abbildung der Programm- und Parameterstruktur von TUSTEP in XML
  - Bewahrung des Funktionsumfangs und der Vernetzbarkeit der Einzelprogramme
  - Umsetzung auch der TUSTEP-internen Programmiersprache TUSCRIPT?
  - Datenübergabe XSTEP -> TUSTEP-Prozesse -> XSTEP
  - Rückgabe von Fehlermeldungen, Zuordnung zu den entsprechenden Skript-Teilen
  - Pattern-Matching:

        - Verhältnis von Regular Expressions zu TUSTEP-Patternmatching klären (am besten sollte beides unterstützt werden)

    - Codierung von Spitzklammern in XSTEP-Skripten (gerade beim Einbringen von Tags in Daten)

    - TUSTEP-Dateiformat

        - Organisation von TUSTEP-Dateien bisher in nummerierten Sätzen (für viele TUSTEP-Funktionen zentral): wie lässt sich das, wo nötig, umsetzen (z.B. Überführung in Tags und PIs)?

        - Beschränkung des Satzlänge auf 64K muss (möglichst für den User unsichtbar) umgangen werden.

    - Eigener Namespace <xstep:

ABGRENZUNG ZU UND INTEGRATION VON TUSCRIPT

- Entscheidung: XSTEP beschränkt sich zunächst auf die Umsetzung der parametergesteuerten TUSTEP-Programme. TUSCRIPT eignet sich nicht im gleichen Maße für die Umsetzung in eine XML-Syntax.

- Für TUSCRIPT ist mittelfristig eine Entwicklungsumgebung (IDE, Extension) in Eclipse erwünscht.

- Die Einbindung von TUSCRIPT in XSTEP soll – analog Javascript in HTML – möglich sein, etwa durch Includieren von externen TUSCRIPTen, evtl. auch durch TUSCRIPT-Sections (CDATA-Sections oder Kommentare) innerhalb von XSTEP.

GRUNDZÜGE VON XSTEP

- Modularer Aufbau
  - Komplexe Abläufe sollen sich aus einzeln austestbaren Einzelskripten zusammensetzen lassen
  - ähnliche Elemente (z.B. Parameterarten) sollten in allen Kontexten im XML gleich strukturiert sein
- Leistungsumfang:
  - unbedingt gebraucht werden:
    - VERGLEICHE, VAUFBEREITE
    - RVORBEREITE, RAUFBEREITE
    - SORTIERE, SVORBEREITE, ggf. SPRUEFE
    - KAUSFUEHRE, EINFUEGE
  - optional: SATZ
  - nicht benötigt werden:
    - FORMULARAUFBEREITE
    - FORMATIERE
    - DRUCKVORBEREITE
- Sonderfall: KOPIERE
  - Problem: Die interne Ablaufsteuerung über Sprungtabellen ist kaum sinnvoll in XML abzubilden.
  - Vorschlag W. Ott: alles beibehalten bis auf mehrere Durchgänge: stattdessen mit IF-THEN-ELSE und CASE, LOOP mehrfach das Modul aufrufen oder die Durchgänge abbilden.
  - Ein stufenweiser Ausbau ist möglich. Die Funktionen von Kopiere sind auch durch TUSCRIPT-Kommandos umsetzbar.
  - Ist ein "Expertenmodus" mit direkter Code-Eingabe als CDATA sinnvoll?
- Möglichkeit zur Einbindung bestehender #tue-Prozeduren als CDATA-Section innerhalb XSTEP oder/und per INCLUDE
- Möglichkeit zur Strukturierung und Ablaufsteuerung innerhalb von XSTEP

- Benutzerschnittstelle realisierbar als Oxygen-Framework:
    - XML-Schema zur Syntaxprüfung und Code-Completion
    - evtl. selbsterklärende Oberfläche auf CSS-Basis in Autoransicht
    - Vorgabe von Standardwerten wo immer sinnvoll: Der User wird auf alle Möglichkeiten hingewiesen, muss aber nur ausfüllen, was er wirklich braucht bzw. von den Voreinstellungen abweicht.
    - Tooltips könnten aus dem Handbuch generiert werden.
- Aus- und Eingabedateien von TUSTEP selbst sollten, wo sie nicht nur intern verwendet werden, im XML-Format stehen:
    - z.B.: Korrektur-Datei aus VERGLEICHE / Korrekturanweisungen für KAUSFUEHRE
- dreistufige Fehlerbehandlung (das XSTEP-Skript muss "am Stück" an TUSTEP übergeben werden und sollte mit XML-Mitteln vorher so weit als möglich validiert sein):
    - Syntaxprüfung bereits bei der Eingabe über XML-Schema (incl. code completion)
    - Nachträgliche Prüfung auf Einhaltung von Konventionen per Schematron
    - Verarbeitung der Fehlermeldungen aus TUSTEP
- Interface-Sprache soll Englisch sein, um einen internationalen Nutzerkreis ansprechen zu können, und zur Integration in die bestehende XML-Welt.
    - Englische Bezeichnungen der Programme und Parameter sind in TUSTEP bereits implementiert, eine Übersetzung des Handbuchs – allerdings auf dem Stand von 1987 erarbeitet und nur bis TUSTEP-Version 1993 aktualisiert – liegt vor.

ERSTER MODELLIERUNGSENTWURF

Ausgehend von der bereits in Blaubeuren und Trier vorgestellten Modellierung von #RVORBEREITE (T.Ott / O. Gasperlin) wurden Fragen der konkreten technischen Umsetzung diskutiert:

- **Modellierungsregeln:** was wird als Element, was als Attribut abgebildet?
    - für eine geschlossene Liste von möglichen Parameterwerten (ggf. mit Standardwert) wird eine Umsetzung als Attribut angestrebt,
    - für freie Eingaben (z.B. Suchstring) sollen Elementwerte verwendet werden. Diese sind nur in engen Grenzen (z.B. Enumeration per Schema) abprüfbar

    - die Parameter sollen nicht sequentiell am Handbuch entlang umgesetzt werden, sondern von einander abhängige Parameter werden zu logischen Einheiten zusammengefasst.

- **Makrostruktur**
    - Verwendet wird ein eigener Namensraum "xstep:"
    - Einzeln austestbare Module müssen zu komplexen Abläufen verknüpfbar sein
    - Wurzelelement ist <xstep:stylesheet>, das ein für sich ablauffähiges Skript umschließt
    - Toplevel-Elemente sind nicht notwendig die eigentlichen TUSTEP-Programme, sondern es wird eine Möglichkeit zur aufgabenspezifischen Gruppierung geben (z.B. RVORBEREITE, SORTIERE und RAUFBEREITE zu einer Einheit "Register")
    - Analog zu XSLT soll die Möglichkeit geschaffen werden, externe Stylesheets einzubinden.
- Mikrostruktur: Parameter
    - Ähnlich strukturierte Parameter sollen in unterschiedlichen Kontexten Tags mit gleichem Namen erhalten (z.B. Austausch-Zeichenfolgen in #RVORBEREITE, #SVORBEREITE, etc.)
- BEISPIEL: Ersetze bei der Eingabe "DORT" durch "<ort region="BW" land="DE">HIER</ort> außer in "DORTSELBST" oder "DORTMUND"

```
<?xml version="1.0" encoding="UTF-8"?>
<xstep:stylesheet xmlns:xstep="http://www.xstep.org/XSTEP">
  <xstep:austausch art="eingabe">
     <xstep:austauscheinheit>
        <xstep:zf typ="search">DORT</xstep:zf>
        <xstep:zf typ="replace">
           <xstep:create_element name="ort">
              <xstep:create_att name="region">BW</xstep:create_att>
              <xstep:create_att name="land">DE</xstep:create_att>
              HIER
           </xstep:create_element>
        </xstep:zf>
     </xstep:austauscheinheit>
     <xstep:ausnahmen>
        <xstep:zf typ="ausn">DORTSELBST</xstep:zf>
        <xstep:zf typ="ausn">DORTMUND</xstep:zf>
```

```
      </xstep:ausnahmen>
    </xstep:austausch>
</xstep:stylesheet>
```

- Für die Parameterarten I-XII sollen in dieser Art Strukturen vorgegeben werden, die in allen betreffenden Programmen einheitlich verwendet werden.
- Möglichkeiten zur Codierung von Spitzklammern beim Einbringen von Tags:
  Am Beispiel von "`<ort region="BW">Tübingen</ort>`":
  - `<ort region="BW">Tübingen </ort>`
    - schwer lesbar, ist in XML ohnehin vorhanden
  - ähnlich wie in XSLT:
    ```
    <xstep:create_element name=ort>
      <xstep:create_attr name="region" value="BW"/>
      Tübingen
    </xstep:create_element>
    ```
    (s. o. Beispiel)
    - sehr "geschwätzig", muss aber in einer XML-Umgebung möglich sein.
  - mit Ersatzzeichen:
    ```
    {ort region="BW"}Tübingen{/ort}
      oder
    [[ort region="BW"]]Tübingen[[/ort]]
    ```
    - Vorteil: gut lesbar, schlanker Code
    - Nachteil: wieder eine "Insellösung", vor allem bei {…} Kollisionen mit TUSTEP-Code an anderer Stelle (Akzente, #SATZ)

OFFENE FRAGEN

- Patternmatching, Verhältnis zu Regular Expressions
- Wie können XPath-Ausdrücke zur Bearbeitung von XML-Dokumenten (Stärke von XSLT, teilweise aber auch schon in TUSTEP implementiert) und sequentielle Verarbeitung von Zeichenketten (Stärke von TUSTEP) kombiniert werden?
- Wie kann das Dateiformat von TUSTEP (vor allem die in vielen Programmen grundlegende Strukturierung in nummerierten Sätzen) in XML überführt bzw. aus XML generiert werden?

- Datenübergabe: Variablen oder (Scratch-)Dateien?
- Kann die die Aufteilung von Datensätzen wegen der Begrenzung der Satzlänge auf 64.000 Zeichen im Hintergrund ablaufen?
- Fehlerbehandlung im Detail: Rückübersetzung des TUSTEP-Ablaufprotokolls in sprechende Fehlermeldungen mit Bezug auf das ursprüngliche XSTEP-Skript
- Rückwärtskompatibilität:
    - bestehender Code sollte auch in dieser Umgebung verwendbar sein (als Section oder Inkludiert)
    - Muss eine bestehende #tue-Datei in ein XSTEP-Skript übersetzbar sein?

NÄCHSTE SCHRITTE

- Exemlarisch für einzelne TUSTEP-Programme Stylesheets ausprogrammieren, Diskussion, Probleme und Lösungen sammeln
- Formalismen für die einzelnen Parameterarten erstellen
- Konzeption der Struktur oberhalb des Einzelprogramms; Bündelung in Module für versch. Aufgaben (Register, Sortieren etc.)?
- Ablaufsteuerung
- Übersetzung ins Englische

ORGANISATORISCHES

- Koordination vorläufig: W. Ott in Absprache mit T. Ott
- Finanzierung: zugesagt sind für 2010 bis zu 10.000,-€ vom ZDV der Universität Tübingen, Prof. Kaletta für ein Teilprojekt
- M. Osthof stünde für Arbeiten am XML-Schema zur Verfügung
- Verteilung der Arbeiten auf die Beteiligten
- eigene Entwürfe zur Diskussion stellen und Anregungen geben

- Geprüft wird die Möglichkeit eines Antrags auf Bundesmittel zur Förderung einer Kooperation zwischen mittelständischen Firmen (<pagina>, DAASI) und Universitäten.

- Kommunikation:
  - Schaffung eine Mailingliste an der Uni Würzburg (inzwischen bereits in Betrieb)
  - WIKI zum Einstellen und gemeinsamen Bearbeiten von Programmentwürfen
- Nahziel: Vorstellung eines Teil-Prototyps bei der ITUG-Tagung 1.-3. September, dazu Vorbereitungstreffen, ggf. per Konferenzschaltung.
- Angestrebt wird, dass bis zum Jahresende die wichtigsten Schemata vorliegen, so dass vor dem Ruhestand von Herrn Schälkle die Umsetzung einer Schnittstelle zu TUSTEP erfolgen kann.
- Projektabschluss wird angestrebt binnen ca. 2 Jahren