

# VNUML Projektpraktikum

Michael Monreal, Tomasz Oliwa

14. Juni 2006

## Abstract

Entstanden im Projektpraktikum 'Simulationen mit User Mode Linux', der vnuml Multiinstaller und VOToN, das VNUML-Old-To-New Programm

## 1 Multi-Installer VNUML 1.5 und VNUML 1.6

Wir haben einen offline Multi-Installer für VNUML 1.5 und VNUML 1.6 entwickelt.

- Mit bisherigem Offline Installer parallele Installation nicht möglich.
- Wir haben einen Installer entwickelt, der dies kann.
- Im folgenden wird vorgestellt, wie er funktioniert.

### 1.1 Installation, was ist neu?

Der VNUML Multi-Installer liegt auf rechnet2 als vnuml-multi.tar.bz2 Datei bzw. als vnuml-multi Ordner.

Enthalten sind:

- 2 Kernel, 2 Filesysteme, notwendige Pakete und Perlmodule für beide VNUMLs.
- Ein cleaner Skript, was vor dem install Skript ausgeführt werden sollte.
- Das install Skript, basierend auf dem alten VNUML Offline Skript.
- Ein Wrapper für die vnumlparser.
- Ein readme.txt Datei, wo alles nochmal erklärt wird.

## 1.2 Vor der Installation, cleaner.sh

*cleaner.sh* beseitigt alte VNUML Dateien um eine reibungslose Installation zu gewährleisten.

*cleaner.sh* ist als root auszuführen.

```
#!/bin/bash
```

```
rm -rf /usr/local/bin/vnumlparser*
rm -rf /usr/local/man/man1/vnumlparser.1.gz
rm -rf /usr/local/lib/vnuml
rm -rf /usr/local/share/vnuml
rm -rf /usr/local/usr
```

```
echo "Please also backup & remove /root/.vnuml"
echo "if it exists from another installation."
```

Sofern alte Simulationen gesichert werden sollen, Backup von /root/.vnuml und danach /root/.vnuml löschen.

## 1.3 Neues install script

*install* installiert VNUML 1.5 und VNUML 1.6.

*install* ist als root auszuführen.

- VNUML 1.5: /usr/local/vnuml-1.5
- VNUML 1.6: /usr/local/vnuml-1.6
- binary für parser: /usr/local/bin

Im Installer wurden die Installationspräfixe als Bashskriptvariablen benutzt.

## 1.4 Neues install script

Im *install* wird nach unserem angepassten ./configure make und make install unser Wrapper für die vnumlparser installiert.

```
if ( test -e $BINARY_DIR); then
cp -f $LOCALPATH/vnumlparser.pl $BINARY_DIR/vnumlparser.pl
CODE=$?
if [ $CODE != 0 ]
then
echo
echo "ERROR $CODE ! Konnte Datei < vnumlparser.pl > nicht
kopieren."
exit 1
```

```
fi
fi
```

Mit einem neuen Flag kann man den jeweiligen Versionsparser starten. z.b.

```
"vnumlparser.pl --vnuml-1_5 -t simple.xml -vB"
```

## 1.5 Wrapper für die vnumlparser

Unser *vnumlparser* benutzt bashskripting und speziell sed, um den jeweiligen richtigen Parser in /usr/local/vnuml-1.x zu starten.

Wie beim Installer hier nur ein interessanter Auszug.

```
for ARGS in "-15" "-16" "--vnuml-1_5" "--vnuml-1_6"; do
  VNUML_ARGS='echo $VNUML_ARGS | sed "s/$ARGS/"'
done

## Start VNUML 1.5.x
if [ "$1" = "-15" -o "$1" = "--vnuml-1_5" ]; then
  echo "Using VNUML 1.5..."
  perl $VNUML15/bin/vnumlparser.pl $VNUML_ARGS
  exit 0
fi
```

Mit sed werden Argumente genommen um \$VNUML\_ARGS anzupassen.

## 1.6 Probleme

VNUML 1.5 funktioniert einwandfrei.

VNUML 1.6 funktioniert bedingt.

Problem der Berechtigungen.

Erste Lösungsmöglichkeit:

- \$ chown vnuml:root /root
- \$ xhost +  
Nicht so schön, da *chown* des root Ordners und wegen *xhost +* hat jeder Zugang zum XServer.

Zweite Lösungsmöglichkeit:

- \$ vnumlparser ..... -u root  
Wird in unserem vnumlparser aktuell gemacht für VNUML1.6 Szenarios.

Bei den Beispielszenarios von VNUML 1.6 kann man die virtuellen Rechner nicht anpingen.

Es liegt am management interface.

```
<vm_mgmt type="none" />
```

Ein Blick in die VNUML Language Referenz enthüllt:  
"None. No management interfaces are configured in any virtual machine. User will need other ways of accessing."

Bei VNUML1.5 war der type implizit auf "private" gesetzt.

Lösung:

```
<vm_mgmt type="private" />
```

Einige VNUML 1.6 Szenarien starten immernoch nicht.

- Das Problem haben nicht nur wir (Mailing Liste).

## 2 VOToN - VNUML-Old-To-New

Ein Programm, welches VNUML 1.5 XML Szenarios in VNUML 1.6 XML Szenarios überführt.

Umgekehrt erscheint es uns nicht sinnvoll, wegen fehlender Features von VNUML 1.5 wäre Informationsverlust die Folge.

Es benutzt XSLT bzw. dessen Parser xsltproc.

- XSLT: Programmiersprache zur Transformation von XML Files.  
<http://www.w3.org/TR/xslt>
- vnuml-1.5.dtd und vnuml-1.6.dtd (Document Type Definition), Strukturdefinition.  
.../vnuml-1.x/share/xml/vnuml/
- voton.xsl hat unsere Transformationsdefinitionen.
- voton.sh ist das ausführbare bash Skript.  
Beispiel: ". /voton.sh tutorial.xml" erzeugt die transformierte Datei "tutorial-1.6.xml"

### 2.1 Wie funktioniert VOToN?

Das Herzstück von VOToN:

```

## Apply transformation
xsltproc -o $OUTPUT-1_6.xml $STYLE $INPUT &> /dev/null
CODE=$?
if [ $CODE != 0 ]; then
    echo "Error while applying transformation."
    exit 1
fi
echo "Input file $INPUT successfully transformed
into $OUTPUT-1_6.xml."

```

Wobei \$OUTPUT eine bearbeitete String Repräsentation des Ausgangsfiles ist, \$STYLE das voton.xsl bezeichnet und \$INPUT das Ausgangsfile darstellt. voton.xsl ist das XML Transformations Sheet, auf das voton.sh zurückgreift. Auszüge aus voton.xsl:

```

<version>1.6</version>
Im version Tag die Versionsnummer auf 1.6 setzen.

<xsl:copy-of select="vnuml/global/simulation_name"/>
Da wird einfach nur kopiert.

<xsl:for-each select="vnuml/global/ssh_key">
    <xsl:if test="@version != 0">
        <ssh_version><xsl:value-of select="@version"/>
        </ssh_version>
    </xsl:if>
</xsl:for-each>

```

Schleife, select ist ein match darstellt mit dem original xml. Eine if Abfrage mit einer expression im Attribut Test, die erfüllt sein muss. Dinge die VOToN also u.A. beherrscht:

- Tag "host\_mapping" entfernt, Infos wandern nach "vm\_mgmt".
- Neuer Tag "ssh\_version", das alte Attribut von "ssh\_key" wurde ein eigenständiges Tag.

Probleme bereitet das "vm\_mgmt" Tag, denn die Werte zu "network", "mask" und "offset" ergeben sich auf dem alten VNUML 1.5 "ip\_offset" Tag, das ist aber schwer zu übernehmen.

Wer es ausprobieren will, VOToN steht als tar.bz2 Datei zum Ausprobieren bereit.

### 3 Zu finden

vnuml-multi auf rechnet2  
VOToN auf rechnet2