Introducing a Toolset for an easy Management of 3GPP Specifications

Thomas Deinlein, Reinhard German, Anatoli Djanatliev
Computer Networks and Communication Systems, Dept. of Computer Science, University of Erlangen-Nürnberg, Germany
Email: {thomas.deinlein,reinhard.german,anatoli.djanatliev}@fau.de

***Abstract*—The standardization process in mobile communications technology, which is carried out by the 3rd Generation Partnership Project (3GPP), is a constantly progressing process. In particular, current technologies, such as LTE (Long Term Evolution) are frequently being expanded and enhanced. Upcoming technologies, such as 5G NR (5G New Radio) are still in an early specification phase and the number of specifications will grow, too. Due to the increasing abundance of specifications and ongoing changes, handling and managing of updates of existing specifications and getting aware of new specifications is quite difficult. This paper introduces a Python toolset for an easy management of a large number of 3GPP specification documents.**

Keywords–3GPP; Mobile Communications; Specification; LTE; 5G NR; Python

I. INTRODUCTION

The task of the 3rd Generation Partnership Project (3GPP) is the worldwide specification of mobile radio technology. About 95% of the world's 7.8 billion mobile subscribers (Q4 2017) use these technologies [1]. Due to steadily growing user numbers, established technologies such as LTE (Long Term Evolution) are being improved and new technologies such as 5G New Radio (5G NR) are being newly developed. Existing specifications must constantly be adapted due to backward compatibility [2] and new specifications have to be created. The specification of mobile communication technologies is therefore a progressing process [3], whereby the amount of documents will increase. It is not a trivial task to manage the abundance of all these different specifications, which is simplified by the toolset presented in this paper. This paper is structured in four sections. In Section II, the problem statement is described in detail. After that, Section III shows, how the different scripts of this toolset work and what they accomplish. Finally, Section IV summarises the extent of this toolset.

II. PROBLEM DESCRIPTION

The 3GPP portal [4] allows downloading of all published 3GPP specifications. The term specification adresses technical specifications and technical reports as well. If the individual specification number is known, downloading is easy and unproblematic. It is possible to type in the number into a specific textfield directly. The search results area shows a glasses button in the last field on the right side, which refers to the download page of the corresponding specification. However, it is often the case that a certain specification refers to another document or other documents. In the rarest cases, it will be sufficient to download (or look into) only one specification. Further specifications have to be downloaded and taken into account then. If there is an interest of studying specifications of one complete series or technology, it is also possible to specify this in the search settings of the 3GPP portal, e.g., listing all 5G NR specifications. At the time of writing of this paper, such a query has already comprised over 390 different documents. The process of downloading these documents is very cumbersome, because it is - besides the individual download of each specification - only possible to download a list of the specifications found in the form of an Excel spreadsheet. In this Excel sheet, the corresponding download page can be opened by clicking on the specification number. A new browser window will open. The versions tab in the new window lists which versions are available for which release. By clicking on the underlined version number, the respective specification can be downloaded as a zip-file, which often contains one or more Microsoft Word doc(x)-files. This download procedure is a very tedious undertaking. If the downloaded Excel sheet contains several hundred specifications, the individual download of all contained documents is extremely time-consuming. Another problem with the Excel sheet is finding out what exactly a specification describes in detail. Usually, only the title of the specification, which is listed in the Excel sheet, indicates whether this specification is relevant for a particular scope or not. However, the exact scope can only be determined by looking into the specification. A keyword search that includes all specification scopes of a certain series or technology would therefore be very advantageous.

There already is a 3GPP toolset available provided by Netovate [5], which allows to search for 3GPP change requests and for written contributions ("TDocs") at meetings. However, the download of the specifications also has to be done manually.

III. USAGE OF THE TOOLSET

The toolset presented in this section allows the downloading and updating of many 3GPP specifications automatically, as well as a keyword search over the scopes of downloaded specifications. A bibTex-file can also be created for certain specifications. The toolset consists of four Python scripts. The programming language Python is freely available at [6]. Further requirements, such as installed Python packages for using the scripts, are shown at the corresponding download page [7]. There is also a detailed description of the usage of the scripts available. In the following, a short overview of the functionality of the toolset will be given.

A. *specificationsHandler.py*

This script automatically downloads specifications and checks if already downloaded specifications have been updated. First, an individual search must be carried out via the 3GPP portal at [4]. The result of this query can then be saved in the form of an Excel spreadsheet. This file must then be copied and both Excel files must be renamed: One file as `LATEST-EXCELFILE.xlsx`, the other file as `REFERENCE-EXCELFILE.xlsx`. Listing 1 shows how to run the script from command line. The script checks if there are specifications in

 ISBN: 978-1-61208-693-4

the LATEST-EXCELFILE.xlsx that are missing in the REFERENCE-EXCELFILE.xlsx. All missing specifications are copied to the REFERENCE-EXCELFILE.xlsx. Additional columns are also created for later version management. Then, downloading procedure of all specifications in the REFERENCE-EXCELFILE.xlsx is divided into several threads. The corresponding download links taken from the REFERENCE-EXCELFILE.xlsx are requested and the most current version number and date are queried and compared with the information in the REFERENCE-EXCELFILE.xlsx. When running this script for the first time, there will not be any version information available in the REFERENCE-EXCELFILE.xlsx. Thus, every specification in the file will be downloaded. After finishing, the REFERENCE-EXCELFILE.xlsx is updated with the downloaded version number and date, which are used with further runnings to check for updates. Note that each specification can have different version numbers for each release. The script always queries the latest version number and date of the last and penultimate release (by passing the parameter *-3* the third to last release is also taken into account). If different versions are identified by comparing the online version number/date and the number/date in the REFERENCE-EXCELFILE.xlsx, a zip-file with the specification is downloaded into the subdirectory /Specifications. If there is only one single doc(x)-file within the zip-file, it is converted into a pdf-file using Microsoft Word (or LibreOffice, when using Linux) and renamed according to the following scheme: SpecNr_Version_Date_Release.pdf. This ensures that the same specifications are not overwritten by different release versions. Converting the doc(x)-files to pdf is done once for easier handling and, second, due to compatibility with the script pdfExtracter.py. There are two further options: By passing the parameter -wx, all doc-files are converted into docx-files (necessary for docxExtracter.py). If there is no conversion needed, the parameter -w can be used. This is the fastest approach to download a large number of specifications. For example, downloading all 5G NR specifications (about 390 at the time of writing this paper) took less than 15 minutes in several runs (depending on internet connection).

LISTING 1. USAGE OF specificationsHandler.py

```
python specificationsHandler.py LATEST-EXCELFILE.xlsx
    ↪ REFERENCE-EXCELFILE.xlsx [ -3 ] [ -w | -wx ]
```

B. pdfExtracter.py and docxExtracter.py

This script allows a keyword search over all scopes of specifications, which are in the folder /Specifications. All 3GPP specifications have the same chapter structure and the chapter *scope* describes what each specification defines in detail. By searching all the scopes it is easier to find relevant specifications for own purposes instead of just taking into account the specification title. After using the previously introduced script, the /Specifications folder should contain many specifications in the pdf/docx format. By starting the script as described in Listing 2 the *scope* of all specifications in the folder /Specifications will be extracted into a txt-file named scopeOfSpecifications.txt. For this purpose, the folder /Specifications is searched for pdf/docx-files, which are then processed by several threads. Each thread extracts the text of the section *Scope*. After all threads are finished, all extracted scopes are written into the mentioned txt-file. The created txt-file makes it possible to search through many different specification scopes (by opening it in a text-editor and using the search function) in order to find out what a particular specification defines in more detail.

LISTING 2. USAGE OF pdfExtracter.py / docxExtracter.py

```
python pdfExtracter.py | docxExtracter.py
```

C. specificationsToBib.py

This script creates a bibTex-file of specifications included in the REFERENCE-EXCELFILE.xlsx for using with LaTeX. For each specification contained in this file a bibTex-entry will be created. How to run this script is shown in Listing 3. The parameter -i is optional. It is possible to mark important specifications within the REFERENCE-EXCELFILE.xlsx. Therefore, it is necessary to fill out the corresponding field in the column *important* with the value *1*. By passing the parameter -i only the marked specifications are taken into account and are written into the OUTPUT-BIBFILE.bib. All generated bibTex-entries are of the type techreport. The individual bibTex-key is created by combining the specification number, the version and the publishing date, e.g., 38.201V15.0.0D03012018. The version number and the date are also transferred into the note-field in every entry. Thus, it is possible to reference same specifications with different version numbers.

LISTING 3. USAGE OF specificationsToBib.py

```
python specificationsToBib.py REFERENCE-EXCELFILE.xlsx
    ↪ OUTPUT-BIBFILE.bib [-i]
```

IV. CONCLUSION

This paper presented a toolset consisting of four Python scripts for easy handling of 3GPP specifications. The script specificationsHandler.py uses an Excel table as starting point, which can be downloaded from the portal [4]. All specifications contained are automatically downloaded, converted to pdf/docx and can be kept up to date. Another script (pdfExtracter.py / docxExtracter.py) extracts the respective section *scope* from the converted specifications into a single txt-file. This allows a simple keyword search over many specifications in only one file. Finally, the script specificationsToBib.py was introduced. Specifications contained in the Excel sheet can be converted into a bibTex-file by running this script.

REFERENCES

[1] E. Dahlman, S. Parkvall, and J. Sköld, Eds., 5G NR: The next generation wireless access technology. Academic Press, 2018, ISBN: 9780128143230.

[2] "About 3rd generation partnership project (3gpp)," 2019, URL: http://www.3gpp.org/about-3gpp/about-3gpp [accessed: 2019-01-28].

[3] "3gpp ongoing releases," 2019, URL: http://www.3gpp.org/specifications/releases [accessed: 2019-01-28].

[4] "3gpp portal," 2019, URL: https://portal.3gpp.org/ [accessed: 2019-01-28].

[5] "Netovate 3gpp tool," 2019, URL: https://netovate.com/3gpp-tools/ [accessed: 2019-01-28].

[6] "Python programming language," 2019, URL: https://www.python.org/ [accessed: 2019-01-28].

[7] "Toolset for an easy management of 3gpp specifications," 2019, URL: https://github.com/cs7org/3GPPtoolset [accessed: 2019-01-28].

 ISBN: 978-1-61208-693-4