Automated Generation of a Book of Abstracts for Conferences that use Indico Platform

Anna Ilina[1][0000−0002−2498−2091] and Igor Pelevanyuk[1,2][0000−0002−4353−493X]

[1] Joint Institute for Nuclear Research, Dubna, Russia
[2] Plekhanov Russian University of Economics, Moscow, Russia

Abstract. Indico is a software which is widely used in High Energy Physics for conferences and events organization. It gives the possibility to provide registration, abstract submission, timetable of events, materials uploading, etc. The creation of a book of abstracts is an important step for many conferences. It provides an overview of the topics which will be presented during the conference. So, participants may be consulted on the topics discussed in every talk.
Indico platform itself provides a possibility to create PDF documents of the book with all authors, abstracts, and affiliations, but the formatting of the book is quite stiff and all corrections and mistakes should be manually found and fixed using the admin panel. The order of abstracts in the book is also confined by options in the Indico system.
So two important requirements appear greater flexibility in terms of formatting and structure of a book, and the ability to check submitted abstracts automatically. The second requirement is quite important too. Authors may call the same affiliation institutes differently, mix different languages in the "authors" field and the abstract text itself, create too short or too long abstracts.
To fulfill the requirements another approach has been chosen: to use the XML representation of all abstracts, check the correctness of each of them, and create Microsoft Word document DOCX based on a template. This approach has been successfully used during the International Symposium on Nuclear Electronics and Computing 2019.

Keywords: Document generation · Book of abstracts · Automated system · DOCX

1 Introduction

Automatic document generation is an important topic especially for systems that process a big amount of data to generate standardized reports or documents. Manual document creation may be tedious and error-prone. And if the data amount is large and document format is not trivial the creation of a document may become too expensive in terms of time. The worst thing is that without some additional efforts there is no way to prove that the document is free of typos or mistakes.

The task of document generation is important for Joint Institute for Nuclear Research(JINR) - International Intergovernmental Organization which unites scientists from areas of nuclear physics, high-energy physics, neutron physics, information technologies, and radiation biology. JINR organizes and hosts more than 40 international conferences and meetings annually. Organization of a conference requires a substantial amount of work related to conference information publication, registration of participants, abstracts collection, time table creation etc. For this purpose, the Indico system is widely used.

Indico is a software started as a European project in 2002. In 2004, CERN adopted it as its own event-management solution and has financed its development since. Today Indico is an Open Source Software available under MIT license. As a tool for conference organization, Indico provides a web page with information about the event, registration form, abstract submission form, survey form, time schedule of the event, links to web pages with other information related to the event, and some other features. With its rich functionality, in the world of high energy physics, Indico became a de facto standard tool for event organization[1].

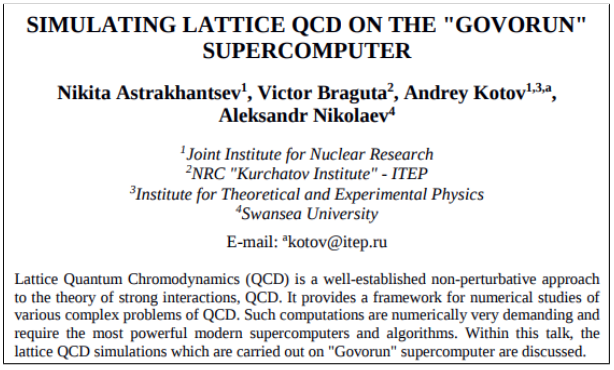
One of the features of Indico is abstract submission form and the possibility to generate the book of abstracts automatically. All abstract texts collected as a plain text, but with the possibility to include LaTeX formulas. The book of abstracts is an important aspect of a scientific conference. It should be accessible before the talks start, so all the participants may get it, get acquainted with it to understand which other talks may be interesting and important to them. The book of abstracts may appear in two forms: a printed copy or a digital copy. It is always possible to read any particular abstract on a page dedicated to a talk if there is no book prepared in Indico.

The task of the creation of a book of abstract is not unique to JINR. But there are not many publications about approaches and methods that allow automating the process. The only example has been found in publication [2]. Authors there use R language and LaTeX to not only generate abstracts but also to generate timetable.

2 Requirements to a Book of Abstracts

Standard abstract consists of several blocks of information: title, list of authors, list of affiliations, corresponding author email, and the text of the abstract itself. Simple example is on the Fig. 1

Each author has one or several affiliations. Affiliations marked by numbers. The email address relates to the corresponding author. Sometimes there are several corresponding authors. Email addresses marked by letters. The Indico system generates rather correct abstract structure. The only issue may be that in Indico generated abstracts the email addresses are displayed, but not connected to a particular author.



SIMULATING LATTICE QCD ON THE "GOVORUN" SUPERCOMPUTER

Nikita Astrakhantsev[1], Victor Braguta[2], Andrey Kotov[1,3,a], Aleksandr Nikolaev[4]

[1]Joint Institute for Nuclear Research
[2]NRC "Kurchatov Institute" - ITEP
[3]Institute for Theoretical and Experimental Physics
[4]Swansea University

E-mail: [a]kotov@itep.ru

Lattice Quantum Chromodynamics (QCD) is a well-established non-perturbative approach to the theory of strong interactions, QCD. It provides a framework for numerical studies of various complex problems of QCD. Such computations are numerically very demanding and require the most powerful modern supercomputers and algorithms. Within this talk, the lattice QCD simulations which are carried out on "Govorun" supercomputer are discussed.

Fig. 1. Simple abstract example.

2.1 Specific Requirements to Book of Abstracts in JINR

Our primary goal was to simplify a process of a book of abstracts creation for the International Symposium on Nuclear Electronics and Computing(NEC) organized by JINR in 2019. Usually, a book of abstract for this conference consists of around 150 abstracts. The creation of this book was performed manually and required a lot of effort. The work was monotonous, boring, and related to many copy/paste cycles. That led to typos and errors during document creation. The work could be divided between several people which lead to spending of around 30 man-hours in total just for one book of abstracts. Manual indexing of affiliations could lead to mistakes and required from the editor an additional check.

In JINR for some conferences, the book of abstracts should be at least as a digital copy in PDF format. And if the printed version is also required JINR Publishing Department accepts PDF documents as a source for books. A simple way is to generate a book using the Indico system. The result will contain the title of the event, list of contents, and all the abstracts sorted by abstract title, presenter name, section name, or some other features. The following requirements make the Indico generated book of abstracts not suitable without some additional editing:

1. After the title page, the second page with the conference annotation in Russian and English language should follow.
2. The third page contains general conference information and topic covered during the conference.
3. The fourth page is a list of Program Committee.
4. The fifth page is a list of Organizing Committee.
5. Then the list of contents follows. But abstracts should be divided by sections and the order of abstracts may be changed in order to bring key talks on top of the section. Various ordering approaches may be used depending on the conference.

6. On one page is only one abstract. Abstracts should be grouped by sections. Section titles should occupy one page, be placed at the center of the page, and be capitalized.

While the first four requirements may be fulfilled by editing a PDF document, requirements 5 and 6 would require substantial efforts in the PDF editor and the list of contents may be easily broken.

Another important issue during the creation of a book of abstracts is the fact that if we just put all of the abstracts from abstract submission forms in one document the inconsistencies and errors will become visible. Among all the different problems the following are more common:

- Authors with the same affiliation writes them differently. For example with the correct "Joint Institute for Nuclear Research" affiliation, we have seen the following variations: JINR, LIT JINR, JINR LIT, Laboratory of Information Technologies JINR, Joint Institute of Nuclear Research, Joint Institute for nuclear research, etc. The biggest issue here is that the affiliation of the author is asked once during the registration of the Indico account. So, the author cannot change affiliation filling the abstract submission form. To do that author should go to the Indico profile settings to change affiliation there.
- Abstract titles in the book of abstracts may be either fully capitalized or just starting with the capital. Authors by themselves decide whether to write it in capital or not. The editor of the book of abstract should check every title manually and bring it to the right format.
- The languages of different pieces of information about abstracts may be written in a different language. For example, during registration, the author wrote the name in Russian, but during abstract form submission used English. The final book created automatically by Indico will use Russian for names and English for abstract text. That is a discrepancy and it should be fixed. Generally, it requires organizers to communicate with the author to get the right spelling of the name in the language of the abstract text.
- Abstract text is usually confined by 250 words. Some authors may violate this rule. The editor sometimes should manually check the number of words and send requests to authors to shorten texts.

2.2 Automatic Processing and Corrections of the Data for a Book of Abstracts

The described requirements and issues demonstrate that the problem of the generation of a document with some specific format is not the only issue with document generation. The analysis and automatic corrections of the text are also possible and required.

It is possible to correct the affiliation name or capitalize on the title of an abstract. But, mixing of languages inside an abstract usually requires organizers to communicate with authors. To perform automatic analysis of texts for book abstracts it is required to have a convenient source of data. Fortunately, Indico

provides the possibility to download the XML document with the representation of all authors and abstracts. That XML document may be used to find usual inconsistencies. After that, the corrections may be done manually in the Indico system. In the newer versions of the Indico, the XML format has been replaced with JSON. In the scope of this article, we will refer only to XML since it was the only option available for use at the moment.

Once the information about all abstracts and necessary corrections are available, it is possible to make all corrections manually and generate a document. Or use that data to generate the final document automatically. This would give great flexibility in terms of content and the form of the final document.

3 Methods of Document Generation

The final goal of document generation is the creation of a PDF file that can be distributed as a digital copy or be printed in a printing house. The only disadvantage of the PDF format is the fact that PDF is relatively difficult to edit. The possibility to edit the generated document is necessary since some manual changes may be required during the later stage of the book creation. The generation of the PDF file directly from the data is also not as simple as a generation of PDF from HTML, DOCX, or TEX formats. We will overview just two common ways to generate PDF: from TEX file using the LaTeX document preparation system, and from DOCX file using Microsoft Word word processor.

3.1 Using the LaTeX System

LaTeX is the standard for the publication of scientific documents. It provides high-quality document printing, so the document looks like a book. The system allows generating a document with specified formatting and the possibility to draw mathematical formulas. In our case, it is possible to draw mathematical formulas directly from abstracts texts.

The biggest advantage of this method is the possibility to generate a TEX file without third-party libraries. Once the template TEX file is available it is rather easy to fill it with text form source XML file. And LaTeX system itself is free software and could be used to generate documents without any payments. And some modern TEX editors even support WYSIWIG mode (What You See Is What You Get, means that editing software allows content to be edited in a form that resembles its appearance when printed or displayed as a finished product), although this type of editors may be non-free.

However, the preparation of a template in the LaTeX system usually requires some special skills. Another issue with LaTeX was the fact that not all organizing committees had a LaTeX template available. Some committees used a complex DOCX template with macros to apply correct formatting to different parts of a text.

3.2 Using a Prepared DOCX Template

This method involves the use of a prepared DOCX template. Templates may be different. It is possible to make it simple and just define several types of formatting for different fields, like "Abstract title", "Authors", "Abstract text". Sometimes more complex templates with macros may be used. The resulting document may be generated from a template using special third-party libraries. The generated DOCX document may be additionally edited in Microsoft Word. The WYSIWIG is originally supported by Microsoft Word and may simplify the manual editing process.

The disadvantage of this approach is the need to use third-party libraries. There is no guarantee that they will always generate the correct document and the newer version of Microsoft Word may render generated DOCX files differently. Microsoft Word itself is non-free software, however, currently, it is a standard software for document creation in many organizations, including JINR. So, everybody has access to Microsoft Word. Another issue is the support of LaTeX formulas. There are some approaches to include them in DOCX documents but most of them require some manual operations.

There were three reasons for us to use DOCX templates for the generation of the book of abstracts. First, the DOCX template has already been created and used for several previous NEC conferences. Second, libraries for Python language for work with DOCX templates and documents have been found, and their functionality was proved during initial tests. Third, in our case, members of the organizing committees responsible for the books of abstracts preferred Microsoft Office.

4 Implementation of the Book of Abstract Generation System

4.1 Tools and Technologies Used for Implementation

The Python3 language has been chosen as a primary language for the developed system. It is possible to use programs developed in Python under Linux and Windows operating systems. Moreover, Python provides the possibility to make the developed program available as a web-application. The Python is quite popular in science organizations and the developed system may be easily used and changed by other users.

The following Python libraries were used in the developed system:

- *xml.etree.cElementTree* is a Python standard library written in C for extracting data from XML files. Now, since Indico system introduced export in JSON format the json library from standard Python libraries may be used.
- *python-docx* is Python library for reading, writing and creating sub documents files [3].

– *python-docx-template* is a library for modifying DOCX documents that have already been created. *python-docx-template* was created because *python-docx* is powerful for creating documents, but not for modifying them. This package itself based on *python-docx* [4].

The developed system uses the Command Line Interface. It has been chosen since it is much easier to make it look the same on both Linux and Windows operating systems. The possible next step is providing a developed system as a service through a Web Interface.

4.2 General Scheme of the Program

The generation of a book of abstracts is done in several steps:

1. Export of the XML file with all abstracts data.
2. Parsing of XML file in Python and creating object with all data from XML file.
3. Automatic correction of standard inconsistencies described in section 2.1.
4. Displaying the notifications about issues that cannot be fixed automatically.
5. Generation of the "preface" part with general conference information described in list of requirements items 1-4 section 2.1.
6. Generation of abstracts part which contain only abstracts.
7. Creation of the final document by concatenating preface and abstract parts.

To allow the execution of these steps several additional files are required: templates, information about the conference to be used for preface generation, CSV file with validated affiliation names.

Preparation of a DOCX Template for Preface. The preface template will determine the look and layout of the preface part of the book. For that, an existing book of abstracts from a previous conference is used. Instead of the information which changes between conferences, we insert variables enclosed in curly brackets. Each variable must have a unique name. If a variable with the same name is repeated in several places, the text associated with it will be inserted instead of all of them. In places where the information may change (like endings of ordinal numbers in conference title) and can not be automatically inserted, the Microsoft Office Word notes used. They show places where the editor's attention is required. Styles should be chosen and applied during template preparation because they will be used in the final document of the book of abstracts. The example of a preface template is shown in Fig. 2.

Making an Additional XML File with Data for Preface part. To fill the previously described template additional XML file should be created. It contains general information about the conference (conference name and number, date, description) in two languages (see Fig. 3). It should be done once for every book of abstracts.

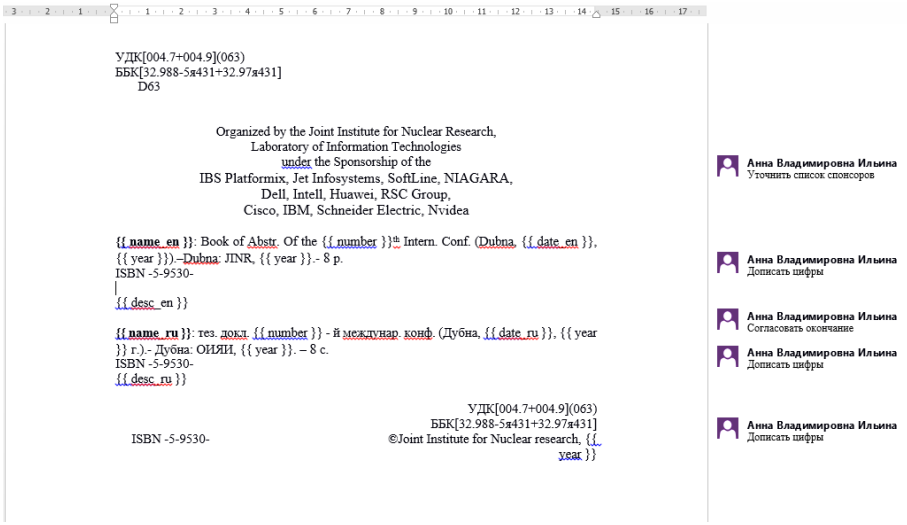


Fig. 2. Example of a DOCX template for preface.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Conference>
    <NameEn>Distributed Computing and Grid-technologies in Science and Education</NameEn>
    <NameRu>Распределенные вычисления и ГРИД-технологии в науке и образовании</NameRu>
    <AdditionalInfoEn>(GRID 2018)</AdditionalInfoEn>
    <AdditionalInfoRu>(ГРИД 2018)</AdditionalInfoRu>
    <Number>8</Number>
    <Year>2018</Year>
    <DateEn>10-14 September</DateEn>
    <DateRu>10-14 сентября</DateRu>
    <DescriptionEn>The book contains abstracts of the reports submitted to the 8th International Con
The main purpose of the Conferenceis to discuss the current Grid operation and the future role of th
The submitted abstracts are described the development of the global grid-infrastructures and such th
    <DescriptionRu>В сборник включены аннотации докладов, представленных на 8-ю международную конфер
Основной целью конференции является обсуждение текущего состояния и будущего развития распределенных
Представленные аннотации описываются развитие глобальной Грид-инфраструктур и бурно развивающиеся те
</Conference>
```

Fig. 3. XML file containing an information about a conference.

Making a CSV File. Since the same affiliations may be written by different authors differently it is important to bring them to consistency at least in one particular book of abstracts. For that purpose, the CSV file is used. It works like a dictionary. All ways to write a particular affiliation should be included in this file. If the name of affiliation used inside an abstract description does not exist in the CSV file, the warning message appears and suggests to add it to the file. If the spelling of the affiliation is correct it still has to be in the CSV file and refer to itself. This ensures that the editor confirms the correction of the affiliation name. An example of a text from the described CSV file is shown in Fig. 4.

```
JINR:Joint Institute for Nuclear Research
LIT JINR:Joint Institute for Nuclear Research
LIT, JINR:Joint Institute for Nuclear Research
JINR LIT:Joint Institute for Nuclear Research
JINR, LIT:Joint Institute for Nuclear Research
FLNR JINR:Joint Institute for Nuclear Research
JINR (Joint Institute For Nuclear Research):Joi
Jinr:Joint Institute for Nuclear Research
jinr:Joint Institute for Nuclear Research
```

Fig. 4. Variety of incorrect Joint Institute for Nuclear Research spellings.

When parsing an input XML file with abstracts, the program accesses CSV file and compares the spelling of the current Affiliation with the correct one. If it matches one of the incorrect variants, the program replaces it with the correct one and inserts it to the final document in an appropriate place. If no such Affiliation is found in the CSV file, the program displays a message stating that this Affiliation is not in the CSV standards file.

Checking the Language Consistency. To ensure that different parts of the same abstract are not written in different languages, we created a special function checking and indicating the language inconsistency (see Fig. 5).

```
More than one language is used in abstract with Id: 180
Email to contact Speaker:  ['aleksey@jinr.ru']
 Speaker's name:  ['Aleksey Bondyakov']
{'Author0_Affiliation': 'LATIN',
 'Author0_Name': 'LATIN',
 'Content': 'CYRILLIC',
 'Title': 'CYRILLIC'}
```

Fig. 5. Program output when different languages are found in an abstract.

Checking the Amount of Words. The program checks the number of words in the abstract. If it is greater or lesser than the specified limit, the warning is displayed in the log of the execution.

4.3 Making corrections

Since many issues determined by the developed system can not be resolved automatically the editor should resolve them manually. There are at least three ways to perform corrections: in the Indico system itself, in some configuration files (now we have two: XML with conference info and CSV with correct affiliations), in the generated document.

Fixing the data in the Indico system is the best way. Usually, it requires some work from authors of the abstract or at least their agreement. All the changes will be included in the generated book of abstracts during the next generations. This works well with inconsistent language use or incorrect abstract text size. If the fix inside Indico is impractical, like with affiliation names, the special configuration file may be introduced to perform corrections during every generation of the particular book of abstracts. This is what has been done with affiliation names.

The fixing data inside the generated document is a viable option, but only when everything else is established and there are no new generations expected. That is because all changes inside could not be used in the next generation and should be done manually again. In real use-case during the creation of a book of abstracts for NEC conference all of the described approaches have been used.

5 Results

The work involved designing and creating an automated abstract book generator that would be identical to the one created by hand. Besides, we had the task to check the source XML file for possible errors listed in this article. As a result, a software product was created that receives four necessary input files:

- an XML file exported from Indico containing abstracts,
- an XML file containing conference information,
- a CSV file containing the Affiliations writing standards,
- a DOCX template that will be used as a base to generate the final document.

As a result, the program creates a final DOCX file with the formatting used in the template. Besides, the program performs checks for possible errors described in this article. Using the information obtained during checks, an editor of the book of abstracts can decide how to correct mistakes: make changes in Indico, ask for a new feature for the book of abstracts generation tool, or change the final document directly.

The applied approach greatly reduced the amount of effort required to generate a document of a book of abstracts. The pursuit to simplify manual indexing

and copying led to introducing automatic checks and corrections. And developed system allowed to generate a document in a format that is known to the end-user.

The developed product is a command-line interface (CLI) application written in Python v3.5 for Linux and Windows. This requires some specific libraries to be installed on the client machine before the use of the system. All source code is available under GNU General Public License v3.0 at [5].

The software product, methods, and approaches described in this article were used during the preparation of the book of abstracts for the NEC'2019 Symposium. The generated book is available at [6].

References

1. Indico Project , https://docs.getindico.io/en/latest/. Last accessed 25 July 2020
2. Lara Lusa, Andrej Blejec, Automated Preparation of the Book of Abstracts for Scientific Conferences using R and LaTeX: Infor Med Slov 2009, 14(1-2), pp. 10–18.
3. python-docx library documentation, https://python-docx.readthedocs.io/en/latest/. Last accessed 25 July 2020
4. python-docx-template library documentation, https://docxtpl.readthedocs.io/en/latest/. Last accessed 25 July 2020
5. Project GitHub repository https://github.com/trnkv/IndicoAbstract. Last accessed 25 July 2020
6. Generated book of abstracts for NEC2019 conference https://indico.jinr.ru/event/738/attachments/4884/6443/NEC_2019_BoA.pdf accessed 25 July 2020